



KERJA PRAKTIK - KI141330

Rancang Bangun Aplikasi *Partner Storelogy*: Sistem Informasi untuk Pengelolaan Informasi Akun, Toko, dan Aplikasi Pihak Ketiga *Storelogy* Menggunakan Kerangka Kerja Laravel

PT. PICODIO

Jl. Galaxi Klampis Asri Selatan VIII No.12, Medokan Semampir, Kec. Sukolilo, Kota SBY, Jawa Timur 60119

Periode: 06 Juli 2020 - 06 Oktober 2020

Oleh:

Mhd Fadly Hasan (05111740000078)

Pembimbing Departemen:

Abdul Munif, S.Kom, M.Sc.

Pembimbing Lapangan:

Yuki Angelia

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

[Halaman ini sengaja dikosongkan]



KERJA PRAKTIK - KI141330

Rancang Bangun Aplikasi *Partner Storelogy*: Sistem Informasi untuk Pengelolaan Informasi Akun, Toko, dan Aplikasi Pihak Ketiga *Storelogy* Menggunakan Kerangka Kerja Laravel

PT. PICODIO

Jl. Galaxi Klampis Asri Selatan VIII No.12, Medokan Semampir, Kec. Sukolilo, Kota SBY, Jawa Timur 60119

Periode: 06 Juli 2020 - 06 Oktober 2020

Oleh:

Mhd Fadly Hasan (05111740000078)

Pembimbing Departemen

Abdul Munif, S.Kom, M.Sc.

Pembimbing Lapangan

Yuki Angelia

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN KERJA PRAKTIK

Rancang Bangun Aplikasi *Partner Storelogy*: Sistem Informasi untuk Pengelolaan Informasi Akun, Toko, dan Aplikasi Pihak Ketiga *Storelogy* Menggunakan Kerangka Kerja Laravel

Oleh: Mhd Fadly Hasan

05111740000078

Mengetahui,
PT PICODIO
Pembimbing kerja praktik



Yuki Angelia

Menyetujui
Dosen Pembimbing Kerja
Praktik



Abdul Munif, S.Kom., M.Sc.
NIP. 19860823 201504 1 004

[Halaman ini sengaja dikosongkan]

Rancang Bangun Aplikasi *Partner Storelogy*: Sistem Informasi untuk Pengelolaan Informasi Akun, Toko, dan Aplikasi Pihak Ketiga *Storelogy* Menggunakan Kerangka Kerja Laravel

Nama Mahasiswa : Mhd Fadly Hasan
NRP : 05111740000078
Departemen : Informatika FTIK-ITS
Pembimbing Departemen : Abdul Munif, S.Kom., M.Sc.
Pembimbing Lapangan : Yuki Angelia

ABSTRAK

Storelogy adalah adalah sebuah e-commerce yang dimiliki oleh PT Davinti Indonesia dan dibentuk oleh PT Picodio, anak perusahaan PT Davinti Indonesia. Untuk meningkatkan layanan, PT Picodio mulai menambahkan fitur dengan membuat subsistem-subsistem baru. Tujuan dari kerja praktik ini adalah membuat subsistem bernama Storelogy Partner yang berupa sebuah Website desktop untuk melakukan proses CRUD yang mengatur aplikasi pihak ketiga, akun, dan anggota tim pengembang aplikasi, serta fitur-fitur pendukung lainnya. Subsistem ini dibangun menggunakan kerangka kerja PHP Laravel, menggunakan basis data MYSQL, CSS Bootstrap dan jQuery sebagai pembentuk antarmuka.

Kata kunci: *Storelogy, Storelogy Partner, Website, CRUD, Laravel, MYSQL, Bootstrap, jQuery.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur saya panjatkan kepada Allah SWT karena berkat karunia-Nyalah penulis dapat menyelesaikan Kerja Praktik yang berjudul: Rancang Bangun Aplikasi *Partner Storelogy*: Sistem Informasi untuk Pengelolaan Informasi Akun, Toko, dan Aplikasi Pihak Ketiga *Storelogy* Menggunakan Kerangka Kerja Laravel.

Penulis menyadari bahwa masih banyak kekurangan dari sistem yang telah dibuat maupun penyusunan buku laporan kerja praktik ini. Penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi untuk kedepannya.

Penulis juga ingin menyampaikan terima kasih kepada pihak-pihak yang telah membantu pembuatan sistem dan buku laporan secara langsung maupun tidak langsung, antara lain:

1. Ayah dan Ibu Penulis.
2. Bapak Abdul Munif, S.Kom., M.Sc. selaku dosen pembimbing kerja praktik selama kerja praktik berlangsung.
3. Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D. Sebagai koordinator kerja praktik
4. Ibu Yuki Angelia dan Bapak Kelvin Allendro Susetyo selaku pembimbing lapangan.

Medan, 22 Oktober 2020

Mhd Fadly Hasan

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN KERJA PRAKTIK	v
ABSTRAK.....	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvii
DAFTAR KODE SUMBER.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan	1
1.3 Manfaat	1
1.4 Rumusan Masalah.....	2
1.5 Lokasi dan Waktu Kerja Praktik.....	2
1.6 Metodologi Kerja Praktik.....	5
1.7 Sistematika Laporan.....	6
BAB II PROFIL PERUSAHAAN.....	9
2.1 Visi dan Misi Perusahaan	9
2.2 Produk Perusahaan	9
BAB III TINJAUAN PUSTAKA	11
3.1 PHP	11
3.2 MySQL.....	11
3.3 Jquery	11
3.4 Bootstrap	12
3.5 Laravel	12

BAB IV ANALISIS DAN PERANCANGAN	13
4.1 Analisis Sistem	13
4.2 Definisi Umum	13
4.3 Analisis Kebutuhan.....	14
4.3.1 Kebutuhan Fungsional.....	14
4.3.2 Kebutuhan Non-Fungsional	15
4.4 Perancangan Sistem	16
4.4.1 F-001 Login Partner	19
4.4.2 F-002 Membuat Akun	20
4.4.3 F-003 Mengubah data diri	21
4.4.4 F-004 Membuat toko	22
4.4.5 F-005 Mendaftarkan Aplikasi	23
4.4.6 F-006 Melihat Toko yang Dibuat.....	24
4.4.7 F-007 Melihat Aplikasi yang dikembangkan.....	25
4.4.8 F-008 Mengubah detail aplikasi.....	26
4.4.9 F-009 Menambahkan Anggota Tim Pengembang..	27
4.4.10 F-010 Mengubah listing Aplikasi.....	28
4.4.11 F-011 Menghapus Anggota Tim Pengembang	29
4.4.12 F-012 Menerima Undangan Tim.....	30
4.4.13 F-013 Mengubah Role Anggota Tim.....	31
4.4.14 F-014 Menghapus Aplikasi yang Dikelola	32
BAB V IMPLEMENTASI.....	33
5.1 Implementasi Model	33
5.1.1 Lapisan Model <i>Store</i>	33
5.1.2 Lapisan Model <i>Store_Owner</i>	33

5.1.3	Lapisan Model <i>User</i>	34
5.1.4	Lapisan Model <i>User_data</i>	34
5.1.5	Lapisan Model <i>App_info</i>	35
5.1.6	Lapisan Model <i>App_status</i>	35
5.1.7	Lapisan Model <i>Category</i>	36
5.1.8	Lapisan Model <i>App_category</i>	36
5.1.9	Lapisan Model <i>App_store</i>	36
5.1.10	Lapisan Model <i>Partners_team</i>	37
5.1.11	Lapisan Model <i>Role</i>	38
5.1.12	Lapisan Model <i>Invite</i>	38
5.1.13	Lapisan Model <i>Oauth_Client</i>	39
5.2	Implementasi Middleware	39
5.2.1	CheckAppDeveloper	39
5.2.2	CheckMembershipTeam.....	40
5.3	Implementasi Lapisan Kontrol	40
5.3.1	Store Controller.....	40
5.3.2	New Store Controller.....	41
5.3.3	Application Controller.....	43
5.3.4	New Application Controller.....	44
5.3.5	Detailed Application Controller	46
5.3.6	Edit Application Controller	46
5.3.7	Edit Application Listing Controller.....	49
5.3.8	Manage Team Controller.....	52
5.3.9	Team Controller	53
5.3.10	Detailed Team Controller	54

5.3.11	Settings Controller	55
5.4	Implementasi Antar Muka Pengguna.....	59
5.4.1	Layout.....	59
5.4.2	Halaman.....	61
BAB VI PENGUJIAN DAN EVALUASI		70
6.1	F-001 Login Partner	71
6.2	F-002 Membuat Akun.....	73
6.3	F-003 Mengubah data diri	74
6.4	F-004 Membuat toko.....	76
6.5	F-005 Mendaftarkan Aplikasi.....	77
6.6	F-006 Melihat Toko yang Dibuat	78
6.7	F-007 Melihat Aplikasi yang dikembangkan	79
6.8	F-008 Mengubah detail aplikasi	80
6.9	F-009 Menambahkan Anggota Tim Pengembang	82
6.10	F-010 Mengubah listing Aplikasi	83
6.11	F-011 Menghapus Anggota Tim Pengembang	85
6.12	F-012 Menerima Undangan Tim	86
6.13	F-013 Mengubah Role anggota tim	87
6.14	F-014 Menghapus Aplikasi yang Dikelola.....	88
BAB VII KESIMPULAN DAN SARAN.....		90
7.1	Kesimpulan.....	91
7.2	Saran	91
DAFTAR PUSTAKA		93
BIODATA PENULIS		95

DAFTAR TABEL

Tabel 1.1 Detail pengerjaan minggu 1 sampai minggu 7	3
Tabel 1.2 Detail pengerjaan minggu 8 sampai minggu 14 (selesai) 4	
Tabel 4.1 Kebutuhan Fungsional	14
Tabel 4.2 Kebutuhan Non-Fungsional	15

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 4.1 Rancangan Sub-Sistem Storelogy	13
Gambar 4.2 Use Case Diagram.....	17
Gambar 4.3 Psychical Data Model.....	18
Gambar 4.4 F-001 Login Partner	19
Gambar 4.5 F-002 Membuat Akun	20
Gambar 4.6 F-003 Mengubah data diri	21
Gambar 4.7 F-004 Membuat toko.....	22
Gambar 4.8 F-005 Mendaftarkan Aplikasi	23
Gambar 4.9 F-006 Melihat Toko yang Dibuat.....	24
Gambar 4.10 F-007 Melihat Aplikasi yang dikembangkan	25
Gambar 4.11 F-008 Mengubah detail aplikasi.....	26
Gambar 4.12 F-009 Menambahkan Anggota Tim Pengembang....	27
Gambar 4.13 F-010 Mengubah listing Aplikasi.....	28
Gambar 4.14 F-011 Menghapus Anggota Tim Pengembang	29
Gambar 4.15 F-012 Menerima Undangan Tim.....	30
Gambar 4.16 F-013 Mengubah Role Anggota Tim	31
Gambar 4.17 F-014 Menghapus Aplikasi yang Dikelola	32
Gambar 5.1 Side bar.....	60
Gambar 5.2 Navbar ketika sudah login	60
Gambar 5.3 Navbar Ketika Belum Login.....	61
Gambar 5.4 Halaman Welcomes.....	62
Gambar 5.5 Halaman Login.....	62
Gambar 5.6 Halaman Register	63
Gambar 5.7 Halaman Applications	64
Gambar 5.8 Halaman New Applications.....	64
Gambar 5.9 Halaman Detailed Applications	65
Gambar 5.10 Halaman Edit Applications	66
Gambar 5.11 Halaman Manage Team Member	67
Gambar 5.12 Halaman Applisting.....	67
Gambar 5.13 Halaman Store.....	68

Gambar 5.14 Halaman New Store.....	69
Gambar 5.15 Halaman Settings	69
Gambar 6.1 Pengujian Login Partner	71
Gambar 6.2 Hasil Pengujian Login Partner	72
Gambar 6.3 Pengujian Membuat Akun	73
Gambar 6.4 Hasil Pengujian Membuat Akun	74
Gambar 6.5 Hasil Pengujian membuat Akun beserta detail akun..	74
Gambar 6.6 Pengujian Mengubah Data Diri.....	75
Gambar 6.7 Hasil Pengujian Mengubah Data Diri.....	75
Gambar 6.8 Pengujian Membuat Toko Baru	76
Gambar 6.9 Hasil Pengujian Membuat Toko Baru	77
Gambar 6.10 Pengujian Mendaftarkan Aplikasi.....	78
Gambar 6.11 Hasil Pengujian Mendaftarkan Aplikasi.....	78
Gambar 6.12 Pengujian Melihat Toko yang Dibuat.....	79
Gambar 6.13 Pengujian Melihat Aplikasi yang Dikembangkan....	80
Gambar 6.14 Pengujian Mengubah Detail Aplikasi.....	81
Gambar 6.15 Hasil Pengujian Mengubah Detail Aplikasi.....	81
Gambar 6.16 Pengujian Menambahkan Anggota Tim Pengembang	82
Gambar 6.17 Hasil Pengujian Menambahkan Anggota Tim Pengembang.....	83
Gambar 6.18 Hasil Pengujian Mengubah Listing Aplikasi	84
Gambar 6.19 Pengujian Menghapus Anggota Tim Pengembang ..	85
Gambar 6.20 Hasil Pengujian Menghapus Anggota Tim Pengembang.....	86
Gambar 6.21 Pengujian Menerima Undangan Tim.....	86
Gambar 6.22 Hasil Pengujian Menerima Undangan Tim.....	87
Gambar 6.23 Pengujian Mengubah Role anggota tim.....	87
Gambar 6.24 Hasil Pengujian Mengubah Role anggota tim.....	88
Gambar 6.25 Pengujian Menghapus Aplikasi yang Dikelola	88

Gambar 6.26 Hasil Pengujian Menghapus Aplikasi yang Dikelola89

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 5.1 Lapisan Model Store.....	33
Kode Sumber 5.2 Lapisan Model Store_Owner	34
Kode Sumber 5.3 Lapisan Model User	34
Kode Sumber 5.4 Lapisan Model User_data	35
Kode Sumber 5.5 Lapisan Model App_info	35
Kode Sumber 5.6 Lapisan Model App_status	36
Kode Sumber 5.7 Lapisan Model Category.....	36
Kode Sumber 5.8 Lapisan Model App_Category	36
Kode Sumber 5.9 Lapisan Model App_store.....	37
Kode Sumber 5.10 Lapisan Model Partners_team	37
Kode Sumber 5.11 Lapisan Model Role	38
Kode Sumber 5.12 Lapisan Model Invite.....	38
Kode Sumber 5.13 Lapisan Model Oauth_client.....	39
Kode Sumber 5.14 Lapisan Middleware CheckAppDeveloper	40
Kode Sumber 5.15 Lapisan Model CheckMembershipTeam	40
Kode Sumber 5.16 Fungsi Index di Store Controller	41
Kode Sumber 5.17 Fungsi Index di New Store Controller	41
Kode Sumber 5.18 Fungsi Create Store di new	43
Kode Sumber 5.19 Fungsi Index di Application Controller	44
Kode Sumber 5.20 Fungsi Index di New Application Controller ..	44
Kode Sumber 5.21 Fungsi Create di Store Controller.....	45
Kode Sumber 5.22 Fungsi Index di Detailed Application Controller	46
Kode Sumber 5.23 Fungsi Index di Edit Application Controller...	47
Kode Sumber 5.24 Fungsi Edit di Edit Application Controller	47
Kode Sumber 5.25 Fungsi Edit App Icon di Edit Application Controller.....	48
Kode Sumber 5.26 Fungsi Delete di Edit Application Controller ..	48
Kode Sumber 5.27 Fungsi Index di Edit Application Listing Controller.....	49

Kode Sumber 5.28 Fungsi Submit dan Edit Application Listing Controller.....	51
Kode Sumber 5.29 Fungsi Index dan Manage Team Controller	52
Kode Sumber 5.30 Fungsi Update Team Member dan Manage Team Controller.....	53
Kode Sumber 5.31 Fungsi Delete Team Member dan Manage Team Controller.....	53
Kode Sumber 5.32 Fungsi Accept Invitation dan Team Controller .	54
Kode Sumber 5.33 Fungsi Create Invite dan Detailed Team Controller	55
Kode Sumber 5.34 Fungsi Delete Invite dan Detailed Team Controller	55
Kode Sumber 5.35 Fungsi Index dan Settings Controller.....	56
Kode Sumber 5.36 Fungsi Update User dan Settings Controller	56
Kode Sumber 5.37 Fungsi Update Password dan Settings Controller	57
Kode Sumber 5.38 Fungsi Update User Contact dan Settings Controller.....	58
Kode Sumber 5.39 Fungsi Update User Address dan Settings Controller.....	59

BAB I

PENDAHULUAN

1.1 Latar Belakang

Storelogy.com adalah sebuah *e-commerce* yang memungkinkan pengguna untuk membuat toko daring dengan mudah. *Storelogy* memberikan *template* website, *editor* website yang sederhana dan *domain* unik, serta memberikan pengguna kemudahan dalam pengelolaan tampilan dan fitur toko.

PT Picodio memutuskan untuk menambahkan fitur pengelolaan toko dengan menambah subsistem baru, seperti *Storelogy App Store* berupa aplikasi web yang dapat membantu pemilik toko untuk menambah *third-party application* untuk toko daring yang telah di-*publish* ke publik, serta *Storelogy Partner* yang berperan sebagai halaman manajemen akun, toko yang dibuat, serta *third-party application* yang dikembangkan. Kerja praktik ini terdiri dari rancang bangun *Storelogy Partner* yang dapat melakukan proses *CRUD* basis data untuk aplikasi, toko, dan akun pengguna, serta fitur pendukung lainnya.

1.2 Tujuan

Tujuan kerja Praktik ini adalah membuat subsistem *Partner Storelogy* untuk pengguna *storelogy*. Serta menyelesaikan kewajiban kuliah 2 SKS matakuliah Kerja Praktik di Departemen Informatika, Institut Teknologi Sepuluh Nopember.

1.3 Manfaat

Manfaat yang dapat diperoleh dengan adanya *Partner Storelogy* adalah:

- Memudahkan pengguna untuk mengatur akun.
- Memberi pengguna kemampuan untuk membuat toko, mendaftarkan *third-party application*, berpartisipasi dalam tim pengembang aplikasi.

- Mempermudah intergrasi aplikasi dengan *Storelogy App Store*.

1.4 Rumusan Masalah

Rumusan masalah dari pengembangan *Partner Storelogy* adalah sebagai berikut:

- Bagaimana rancangan fungsional sub-sistem *Storelogy Partner*
- Data apa saja yang dibutuhkan untuk proses *CRUD* di *Storelogy Partner*.
- Bagaimana cara menghubungkan *Storelogy Partner* dengan Subsistem *Storelogy App Store*.

1.5 Lokasi dan Waktu Kerja Praktik


Kerja praktik dilaksanakan pada waktu dan tempat sebagai berikut:


Lokasi	: <i>Work from Home</i>
Alamat	: Jalan Pinang Baris Elok No.1, Medan-Sungai, Medan, Sumatera Utara
Waktu	: 6 Juli 2020 – 06 Oktober 2020 (3 Bulan)
Hari Kerja	: Senin-Jumat
Jam Kerja	: 09.00 WIB - 17.00 WIB

Tabel 1.1 dan Tabel 1.2 adalah rincian jadwal mingguan dari pengerjaan kerja praktik yang telah dilaksanakan.

Tabel 1.1 Detail pengerjaan minggu 1 sampai minggu 7


Kegiatan	M1	M2	M3	M4	M5	M6	M7
Pengenalan Perusahaan dan sistem yang akan dibuat							
Analisis kebutuhan							
Perancangan basis data							
Perancangan <i>workflow</i> atau cara kerja sistem							
Integrasi template Tampilan							
Pengerjaan Fitur teams							
Pengerjaan Fitur partners							
Pengerjaan Fitur mengundang anggota							
Implementasi Laravel Passport							
Persiapan staging repository							
Memperbaiki Bug							
Melanjutkan implementasi Laravel passport							


 = Minggu kerja efektif

 = Libur nasional

Tabel 1.2 Detail pengerjaan minggu 8 sampai minggu 14 (selesai)

Kegiatan	M8	M9	M10	M11	M12	M13	M14
Revisi database							
Pengerjaan Middleware							
Pengerjaan Fitur Store							
Libur 17 Agustus dan Tahun Baru Hijriyah							
Revisi Tampilan Store							
Implementasi Laravel Validate ke semua Form							
Perbaiki routing dan middle ware							
Testing di staging							
Pengerjaan Fitur applisting							
Persiapan hand over ke pekerja lain							
Hand-over ke pekerja lain							
Rapat dan diskusi fitur apa saja yang sudah dikerjakan							

 = Minggu kerja efektif

 = Libur nasional

1.6 Metodologi Kerja Praktik

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

1. Perumusan Masalah

Untuk mengetahui permasalahan apa yang harus diselesaikan, penulis mengikuti rapat harian yang dilaksanakan tiap hari kerja jam 09.00 WIB. Rapat ini membahas progres yang telah dikerjakan karyawan di hari sebelumnya dan memaparkan masalah, saran, atau kekurangan dari progres yang dikerjakan. Dari hasil rapat tersebut dapat ditemukan kendala pada sistem, apabila kendala yang ditemukan cukup rumit, akan diadakan rapat tambahan untuk pihak-pihak terkait untuk membahas solusi dari kendala tersebut. Selain rapat dengan karyawan, penulis, mahasiswa magang lainnya, beserta CEO PT *Picodio* melakukan uji banding dengan sistem serupa untuk mendapatkan gambaran sistem dan merumuskan fitur-fitur apa saja yang akan diterapkan.

2. Studi literatur

Setelah dilakukan pemahaman atas cara kerja *Storelogy*, dilakukan studi lebih mendalam pada teknologi yang akan digunakan untuk implementasi, serta mencari tahu dokumentasi dari teknologi dan fitur yang akan dipakai. Informasi – informasi ini dapat ditemukan melalui internet.

3. Analisis dan perancangan

Tahap berikut merupakan tahap lanjutan berupa analisis fungsional dari subsistem *Storelogy Partner* sehingga menghasilkan daftar *Use case*. Kemudian dari *Use case* tersebut, dapat diperoleh rancangan model, rancangan basis data, dan rancangan konteks.

4. Implementasi

Implementasi adalah tahapan pengerjaan dari rancangan yang dibuat untuk menghasilkan sistem yang diinginkan. Untuk memudahkan dan mempercepat pengerjaan, penulis dan PT Picodio setuju untuk menerapkan Template *bootstrap*. Pengerjaan dari tahap implementasi dilaporkan pada rapat harian.

5. Pengujian dan Evaluasi

Tahapan pengujian dilakukan dengan cara menjalankan tiap *Use case* dan pengecekan apakah sudah sesuai oleh penulis dan CEO. Jika ditemukan kekurangan, akan dilakukan perbaikan dan rapat apabila dibutuhkan.

6. Kesimpulan dan Saran

Akan diberikan kesimpulan dan saran yang didapat oleh penulis selama kerja praktik untuk pembaca.

1.7 Sistematika Laporan

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut:

1. Bab I Pendahuluan

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan, serta sistematika pengerjaan kerja praktik dan juga penulisan laporan kerja praktik.

2. Bab II Profil Perusahaan

Pada bab ini, dijelaskan secara rinci tentang profil perusahaan tempat melaksanakan kerja praktik, yakni PT Picodio Surabaya.

3. Bab III Tinjauan Pustaka

Pada bab ini, dijelaskan mengenai tinjauan pustaka dan literatur yang digunakan dalam pengerjaan sub-sistem selama kerja praktik.

4. Bab IV Analisis dan Perancangan

Pada bab ini, dijelaskan hasil pembelajaran atau analisis terhadap apa saja yang diperlukan dan harus diperhatikan dalam pengembangan sistem yang dikerjakan selama kerja praktik.

5. Bab V Implementasi

Pada bab ini, berisi penjelasan hasil implementasi dari sub-sistem *Storelogy Partners*.

6. Bab VI Pengujian dan Evaluasi

Pada Bab ini, dijelaskan hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik

7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

PT.Picodio adalah salah satu anak perusahaan Davinti Group yang bergerak di bidang *web & mobile apps agency*. PT.Picodio memberikan pelayanan berupa pembentukan *web* atau aplikasi ponsel untuk membantu penjualan bisnis pengguna, serta menawarkan konsultasi strategi pendekatan calon pelanggan yang sesuai agar menaikkan jumlah trafik kunjungan website.

2.1 Visi dan Misi Perusahaan

a. Visi:

“Kami berkeinginan untuk menjadi perusahaan grup digital terbaik di Indonesia dan juga Asia Tenggara.”

b. Misi:

“Kami berkomitmen dalam hal memberikan pelayanan yang terbaik untuk klien tercinta kami.”

2.2 Produk Perusahaan

- *Genesis Camera*: website *e-commerce* aksesoris kamera
- *Storelogy.com*: website *e-commerce* PT Davinti
- *Onegalaxysuperblock.com*: website profil Apartemen *onegalaxy*
- *Catherinwong.com*: website *e-commerce fashion* oleh Catherin Wong
- *Profira-clinic.com*: website profil perusahaan klinik perawatan kulit profira
- *Evancards.com*: website profil perusahaan evan cards
- *Natural Nest*: aplikasi ponsel informasi dan produk sarang burung
- *Interbrucke.com*: website profil perusahaan Interbrucke Group

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses kerja praktik.

3.1 PHP

PHP adalah bahasa pemrograman yang dapat dipakai untuk pengembangan situs web. Bahasa ini dapat mengurus logika dan automasi data yang terjadi di lapisan *back-end* atau server web [1]. Bahasa PHP sangatlah fleksibel, sistem dapat memberikan pengguna HTML atau dokumen PDF, foto, dan video sesuai kebutuhan. PHP juga dapat diintegrasikan dengan basis data populer seperti MySQL, PostgreSQL, dan lain-lain, sehingga situs web yang dibuat lebih dinamis. [2]

3.2 MySQL

MySQL adalah sistem manajemen basis data SQL bersifat *open source* dan dikembangkan oleh Oracle. Basis data MySQL memungkinkan sistem untuk menyimpan data dari pengguna yang akan dibutuhkan oleh sistem. MySQL dapat menampung data yang sangat besar dan dapat melakukan operasi komputasi pada data tersebut dengan cepat. MySQL memiliki sifat basis data yang relational, maksudnya tiap data dapat dipecah menjadi tabel-tabel basis data yang terpisah dalam bentuk model yang memiliki kolom dan baris, tiap tabel-tabel yang terpisah ini bisa memiliki hubungan antar tabel seperti *one-to-one*, *one-to-many*, serta untuk kolom dapat memiliki sifat *unique*, *required* atau *optional*. [3]

3.3 JQuery

JQuery adalah *library* JavaScript yang digunakan di dokument HTML. JQuery memudahkan pemrograman JavaScript tampilan dengan menerapkan *event handling*, manipulasi komponen,

animasi dan AJAX. Dengan fitur tersebut, JQuery dapat mengatur dan memanipulasi tiap komponen HTML sehingga dapat membuat tampilan yang lebih menarik dan responsif. [4]

3.4 Bootstrap

Bootstrap adalah kerangka kerja CSS dan HTML yang digunakan untuk pengembangan tampilan situs web. Kerangka kerja ini memberikan tampilan yang menarik untuk tombol, tulisan, navigasi. Bootstrap memudahkan dan mempercepat pengerjaan tampilan dibandingkan menggunakan CSS dan JavaScript murni. Penggunaan bootstrap disajikan dalam preset atau kelas yang dapat digunakan komponen HTML untuk mendapatkan tampilan yang dibutuhkan. [5]

3.5 Laravel

Laravel adalah kerangka kerja PHP yang mengatur web server, menggunakan *template engine* blade, dan memiliki *design pattern* *Model View Controller*. Kerangka kerja Laravel merupakan salah satu yang paling populer karena cepat, modular, mudah dipelajari, serta memiliki penamaan yang mudah dipahami. Laravel memiliki banyak *library* yang terpisah menjadi komponen-komponen sehingga memudahkan implementasi dan penggunaan. Laravel juga memiliki *routing* yang mengatur rute atau alamat web yang diakses. Laravel juga memberikan fitur ORM yang memungkinkan pengaksesan basis data dalam bentuk objek sehingga memudahkan pengerjaan dan implementasi. [6]

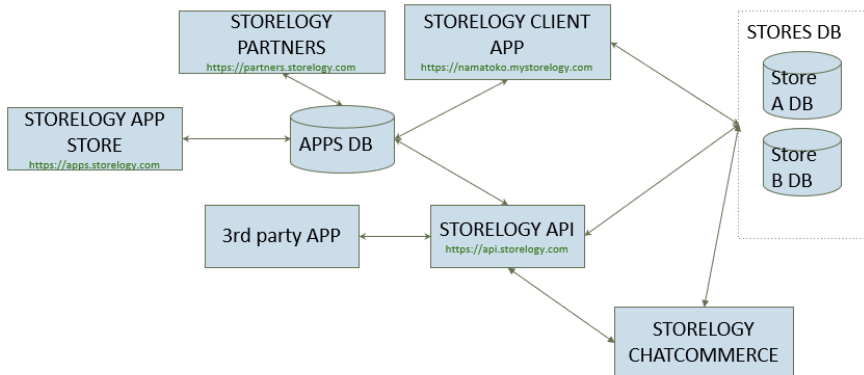
BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis Sistem

Bab ini akan menjabarkan tahapan dalam pembangunan *Storelogy Partner* yang berupa analisis dari sistem yang dibangun. Penjabaran analisis dijelaskan dalam dua bagian, definisi umum dan analisis kebutuhan.

4.2 Definisi Umum



Gambar 4.1 Rancangan Sub-Sistem Storelogy

Sistem pada *Storelogy.com* dibagi menjadi beberapa subsistem, berupa *client App*, *ChatCommerce*, *Partners*, dan *App Store* seperti yang ditunjukkan pada gambar 4.1. Pada Kerja Praktik ini, akan dikerjakan *Storelogy Partners* mulai dari analisis, perancangan, dan implementasi.

Storelogy Partners adalah *website* pengelolaan akun pengguna yang dapat mengelola *Third-party Application* (*3rd party App*) beserta tim pengembangannya, membuat toko baru, dan fitur fungsional lainnya. Subsistem ini akan memiliki satu tampilan

untuk user yang berisi menu untuk *login*, registrasi, mengatur dan menampilkan akun, toko, serta *third-party app* yang dikembangkan diikuti tim pengembangnya.

Third-Party App adalah aplikasi buatan yang dikembangkan oleh pihak diluar *Storelogy* yang dapat dipasang dan digunakan di toko *Storelogy* pengguna. Tiap aplikasi pihak ketiga akan dikembangkan oleh satu tim pengembang baru. Satu aplikasi hanya memiliki satu tim pengembang dan pembuat aplikasi diberikan pilihan untuk mem-*publish* aplikasi tersebut. Apabila aplikasi tersebut merupakan aplikasi publik, aplikasi akan terdaftar di *Storelogy App Store* untuk direview pihak admin *Storelogy* yang dimana akan masuk ke dalam *App Store* agar dapat di-instal oleh toko lain.

4.3 Analisis Kebutuhan

Dalam aplikasi ini, terdapat kebutuhan-kebutuhan yang harus dipenuhi oleh sistem. Kebutuhan-kebutuhan yang harus dipenuhi didapatkan dari uji banding sistem serupa serta diskusi dengan CEO dan mahasiswa magang lainnya. Kebutuhan-kebutuhan ini terbagi ke dalam dua jenis, yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan bagaimana *Storelogy Partners* bekerja untuk memenuhi tujuan yang ditetapkan. Kebutuhan-kebutuhan ini dijelaskan pada Tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional

Kode	Kebutuhan
F-001	Login Partner
F-002	Membuat Akun
F-003	Mengubah data diri

F-004	Membuat toko
F-005	Mendaftarkan Aplikasi
F-006	Melihat Toko yang Dibuat
F-007	Melihat Aplikasi yang dikembangkan
F-008	Mengubah detail aplikasi
F-009	Menambahkan anggota tim pengembang
F-010	Mengubah <i>listing</i> Aplikasi
F-011	Menghapus Anggota Tim Pengembang
F-012	Menerima Undangan Tim
F-013	Mengubah <i>Role</i> anggota tim
F-014	Menghapus aplikasi yang dikelola

4.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan untuk menjelaskan batasan, karakteristik, dan cara kerja dari sebuah sistem yang harus dipenuhi. Kebutuhan-kebutuhan non-fungsional dijelaskan pada Tabel 4.2.

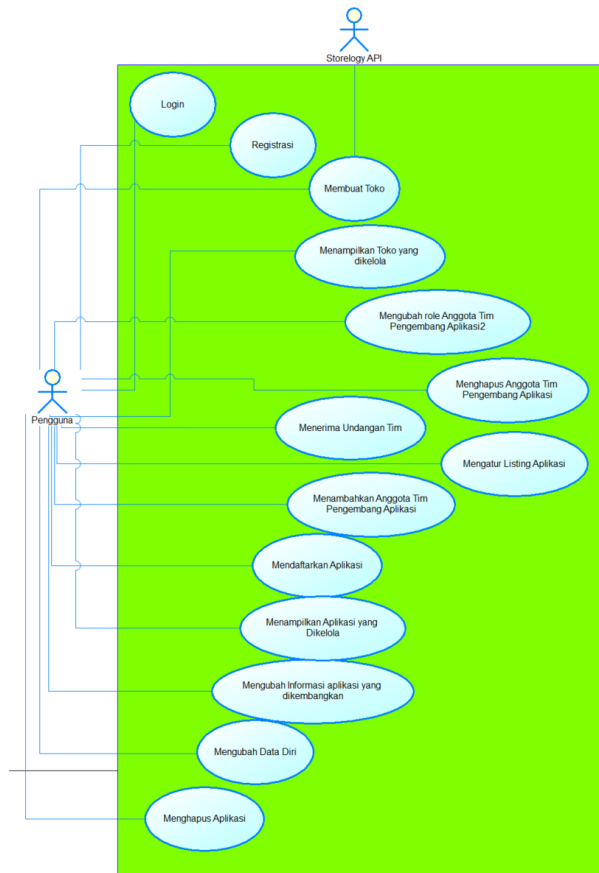
Tabel 4.2 Kebutuhan Non-Fungsional

Kode	Kebutuhan
NF-001	Dapat diakses melalui <i>browser web</i>
NF-002	Memiliki tampilan yang sederhana dan menarik
NF-003	Sistem menunjukkan <i>feedback</i> apabila terdapat kesalahan atau kekurangan saat memasukkan data
NF-004	Menggunakan Bahasa Inggris
NF-005	Pengguna tidak dapat mengakses halaman aplikasi yang tidak diizinkan

NF-006	Pengguna tidak dapat mengakses halaman tim yang bukan ia kembangkan
NF-007	Dapat diakses 24 Jam

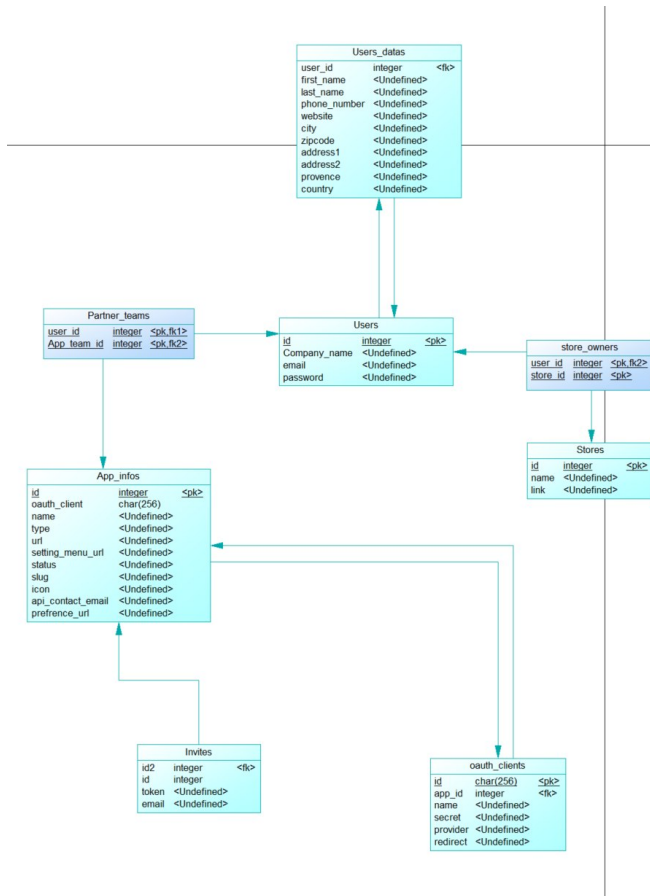
4.4 Perancangan Sistem

Perancangan sistem dilakukan untuk menetapkan proses dan cara kerja pada *Partner Storelogy* sehingga memudahkan pengembangan dan pemeliharaan aplikasi. Bentuk Perancangan sistem yang digunakan adalah *Use Case Diagram*, *Psychical Data Model*, dan *Activity Diagram*.



Gambar 4.2 Use Case Diagram

Gambar 4.2 adalah *Use Case Diagram* yang menunjukkan hubungan dan fungsi antara pengguna, *Storelogy Partners* dan *Storelogy API*.

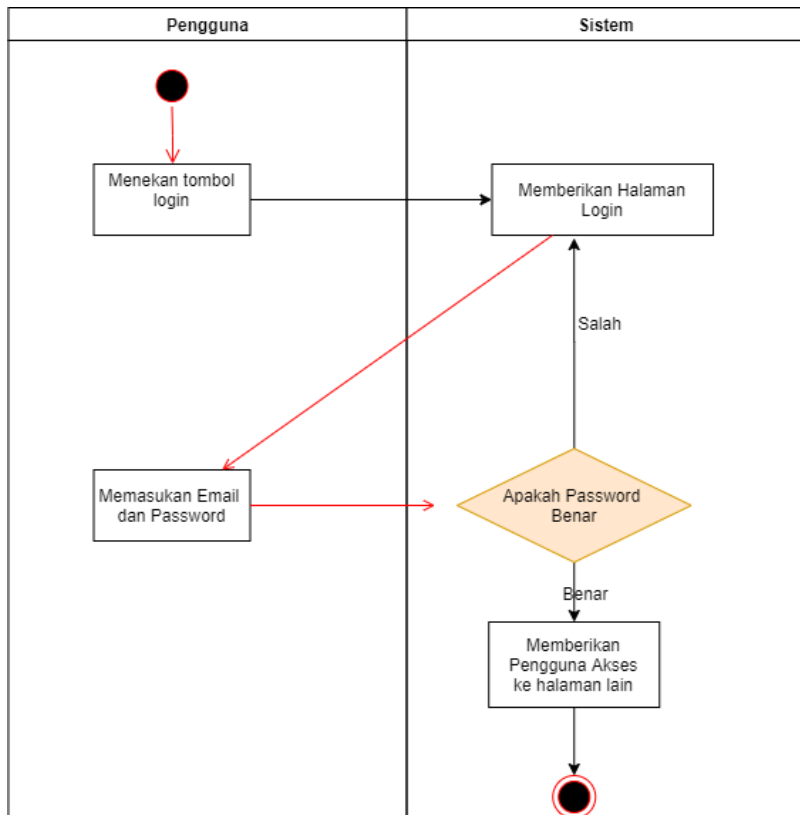


Gambar 4.3 Psychical Data Model

Gambar 4.3 adalah *Physical Data Model* yang menunjukkan gambaran hubungan antar-tabel di basis data *Storelogy Partners*.

4.4.1 F-001 Login Partner

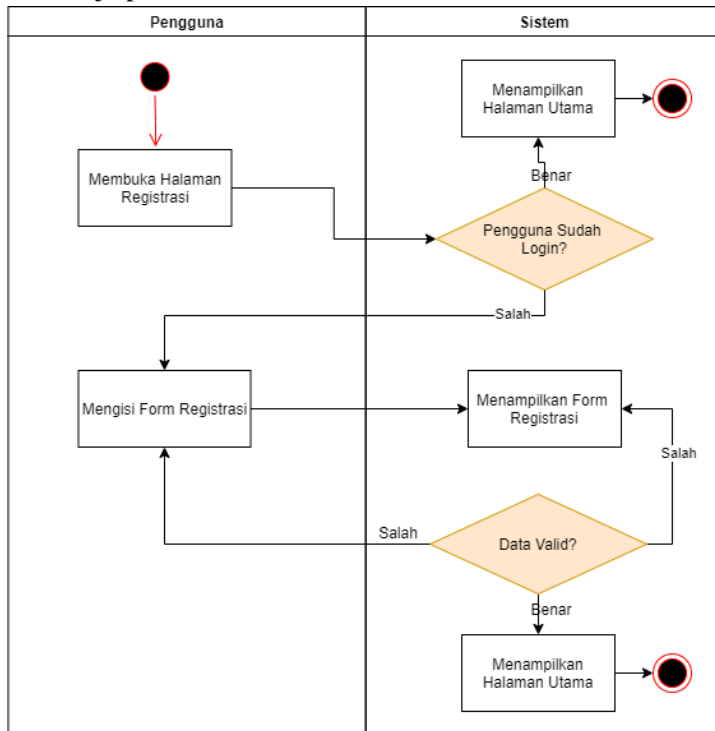
Pengguna dapat *login* ke dalam sistem agar dapat mengakses penuh *Storelogy Partner*. Gambar 4.4 merupakan diagram aktifitas yang menunjukkan alur kerja *Login*.



Gambar 4.4 F-001 Login Partner

4.4.2 F-002 Membuat Akun

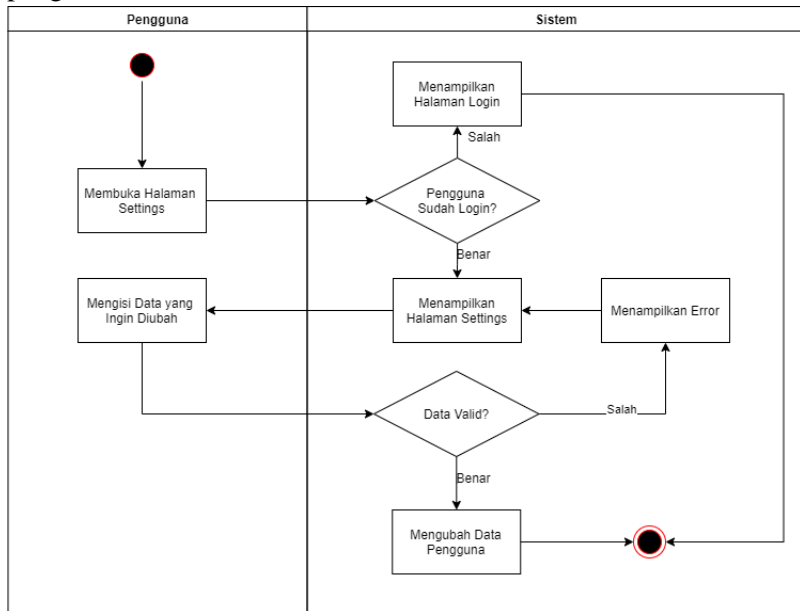
Pengguna dapat membuat akun baru yang dapat dipakai untuk *login* ke dalam sistem. Pengguna akan diminta untuk memberikan informasi diri dan *e-mail* unik. Gambar 4.5 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja pembuatan akun baru.



Gambar 4.5 F-002 Membuat Akun

4.4.3 F-003 Mengubah data diri

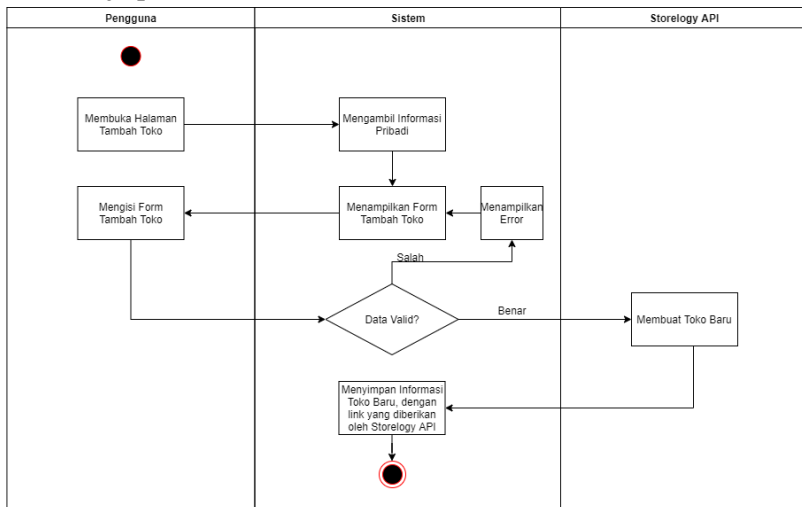
Pengguna yang sudah *login* dapat mengubah data diri yang terdaftar di basis data. Gambar 4.6 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja pengubahan data diri akun.



Gambar 4.6 F-003 Mengubah data diri

4.4.4 F-004 Membuat toko

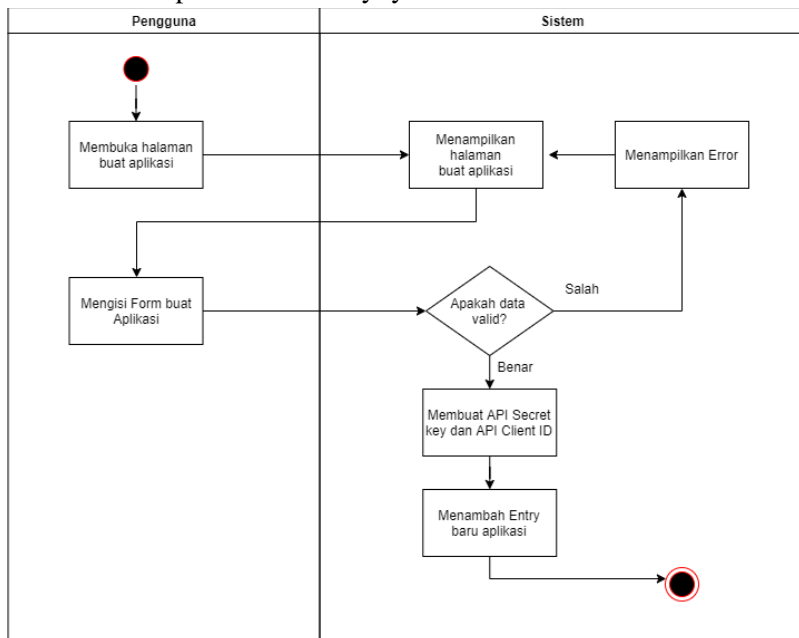
Pengguna dapat membuat sebuah toko baru dengan mengisi informasi-informasi untuk pendaftaran toko, beberapa kolom informasi akan diisi otomatis sesuai dengan data diri pengguna. Informasi-informasi tersebut akan dikirimkan ke *Storelogy API* dimana storelogy api akan membuat toko dan memberikan alamat toko yang dapat diakses pembuat toko. Gambar 4.7 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja pembuatan toko baru.



Gambar 4.7 F-004 Membuat toko

4.4.5 F-005 Mendaftarkan Aplikasi

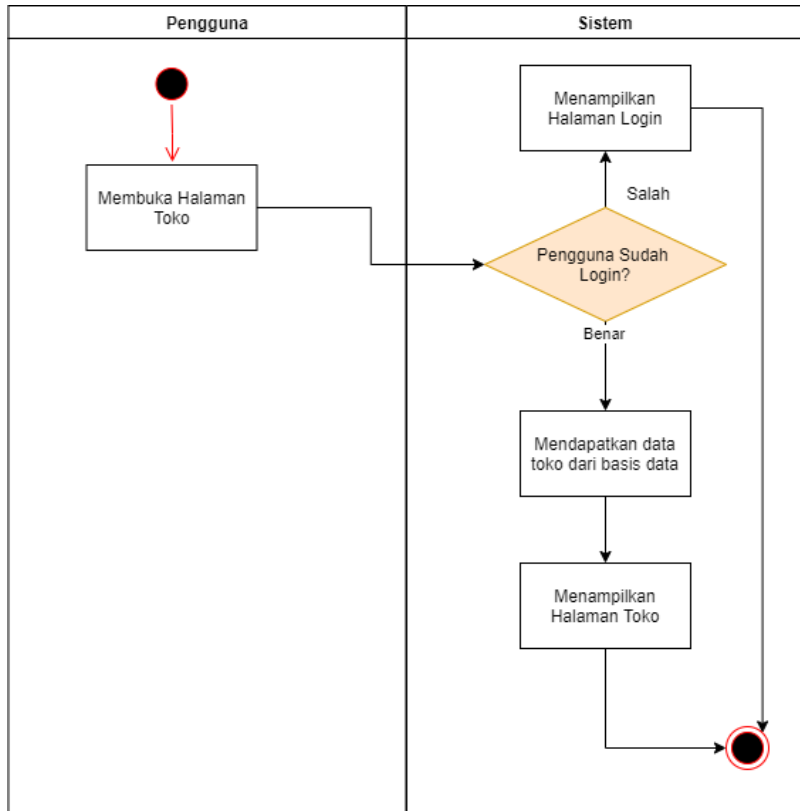
Pengguna dapat membuat sebuah aplikasi pihak ketiga baru dengan mengisi informasi-informasi untuk pendaftaran aplikasi, aplikasi yang dibuat akan diberikan *key Oauth* berupa *client ID* dan *secret ID*. Gambar 4.8 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja pendaftaran aplikasi baru beserta mendapatkan oauth keynya.



Gambar 4.8 F-005 Mendaftarkan Aplikasi

4.4.6 F-006 Melihat Toko yang Dibuat

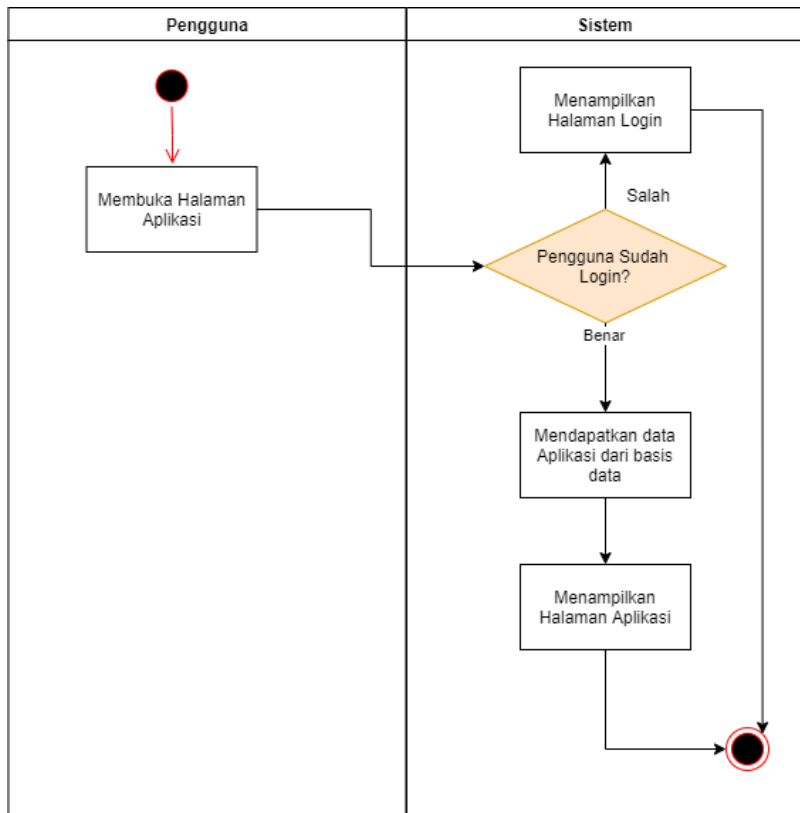
Pengguna dapat melihat toko yang dikelola beserta informasi-informasi tambahan lainnya. Gambar 4.9 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja melihat toko yang dibuat.



Gambar 4.9 F-006 Melihat Toko yang Dibuat

4.4.7 F-007 Melihat Aplikasi yang dikembangkan

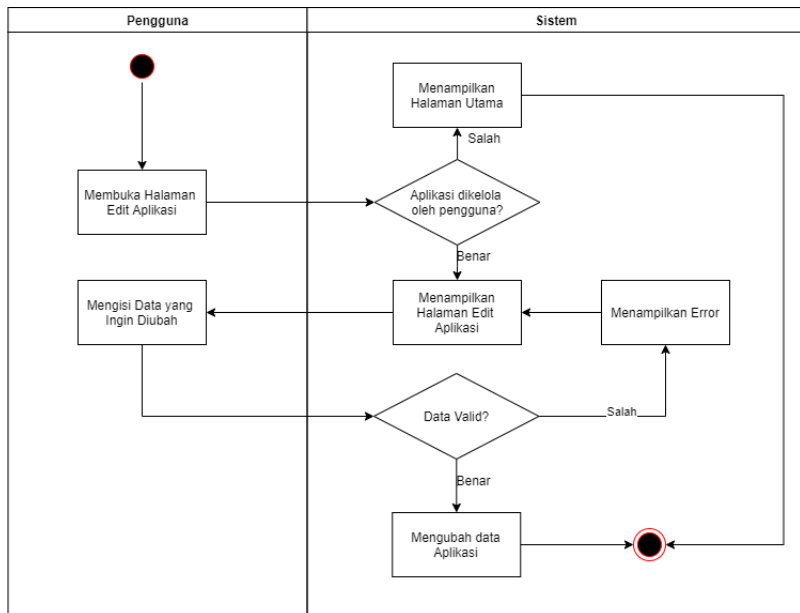
Pengguna dapat melihat aplikasi yang dikelola beserta informasi-informasi tambahan lainnya. Gambar 4.10 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja untuk menampilkan kumpulan aplikasi yang dikembangkan.



Gambar 4.10 F-007 Melihat Aplikasi yang dikembangkan

4.4.8 F-008 Mengubah detail aplikasi

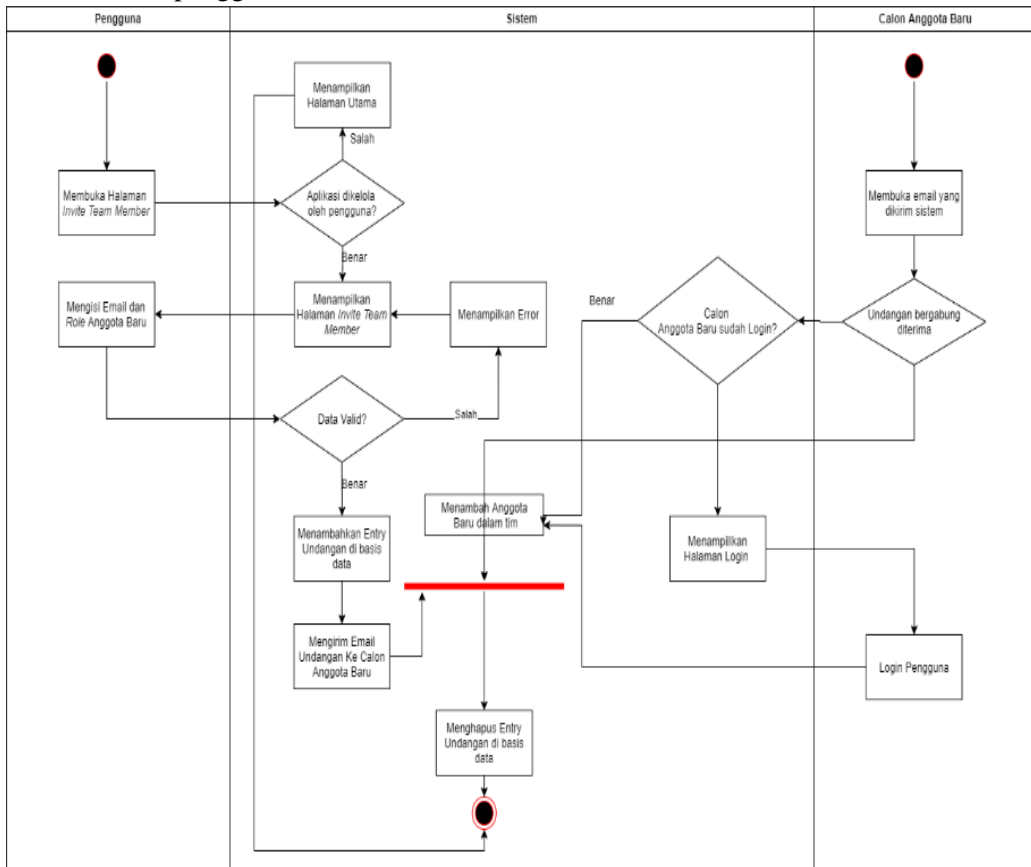
Pengguna dapat mengubah informasi aplikasi yang dikelola. Gambar 4.11 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur perubahan data detail aplikasi.



Gambar 4.11 F-008 Mengubah detail aplikasi

4.4.9 F-009 Menambahkan Anggota Tim Pengembang

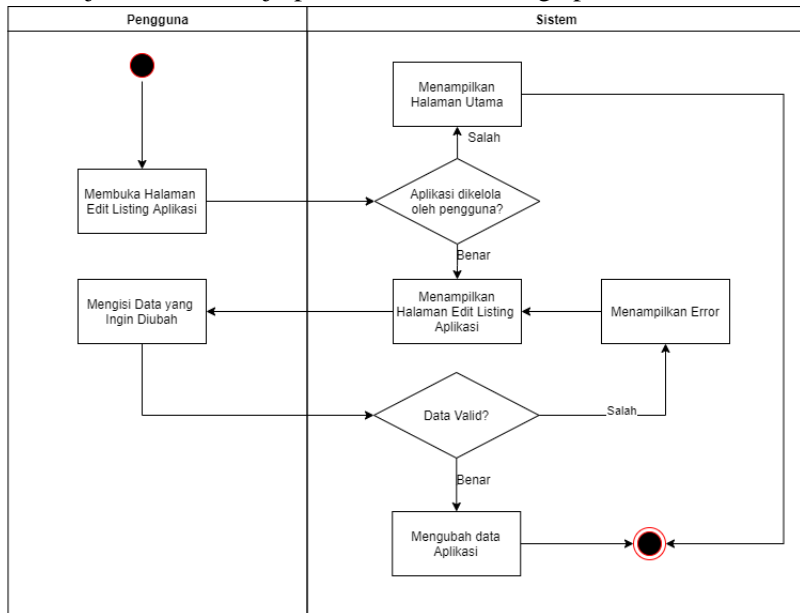
Pengguna dapat menambahkan anggota tim pengembang dan orang yang diajak dapat melakukan konfirmasi. Gambar 4.12 merupakan diagram aktifitas antara pengguna, calon pengguna baru, dan sistem yang menunjukkan alur kerja penambahan keanggotaan calon pengguna baru.



Gambar 4.12 F-009 Menambahkan Anggota Tim Pengembang

4.4.10 F-010 Mengubah listing Aplikasi

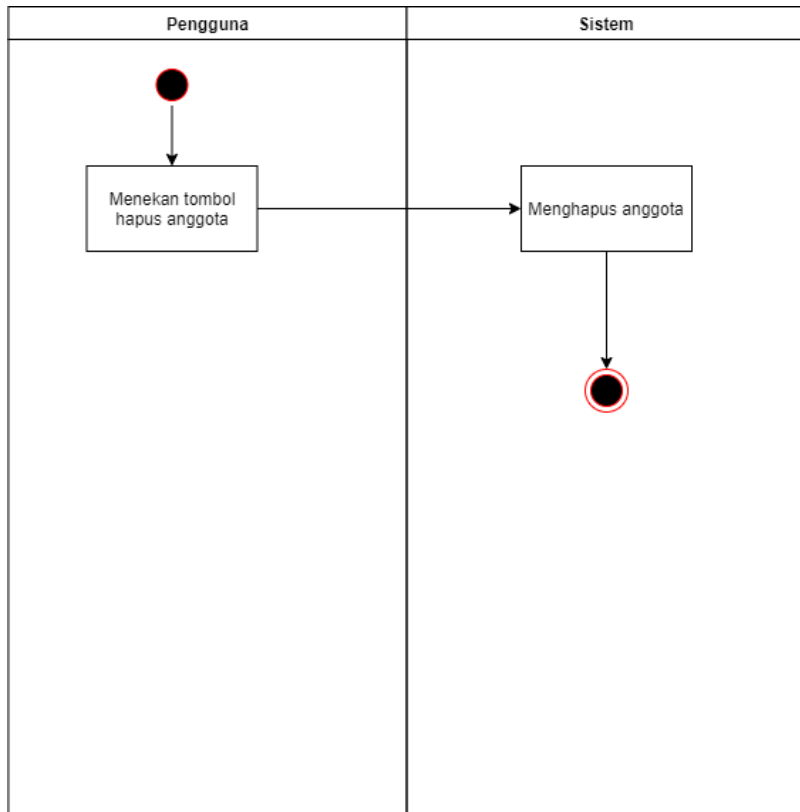
Pengguna dapat mengubah listing aplikasi publik yang dikelola beserta informasi tambahan lainnya. Gambar 4.13 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja perubahan data listing aplikasi.



Gambar 4.13 F-010 Mengubah listing Aplikasi

4.4.11 F-011 Menghapus Anggota Tim Pengembang

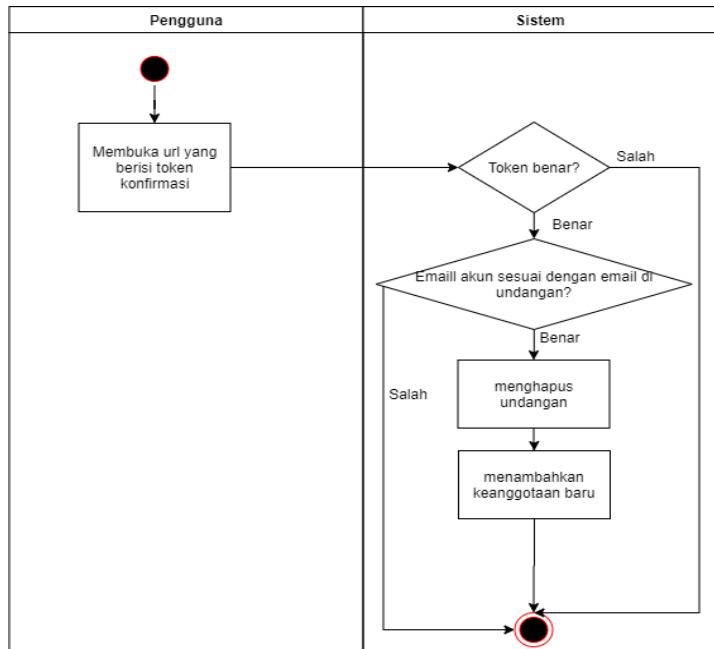
Pengguna dapat menghapus keanggotaan tim apabila dibutuhkan. Gambar 4.14 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur penghapusan keanggotaan tim pengembang.



Gambar 4.14 F-011 Menghapus Anggota Tim Pengembang

4.4.12 F-012 Menerima Undangan Tim

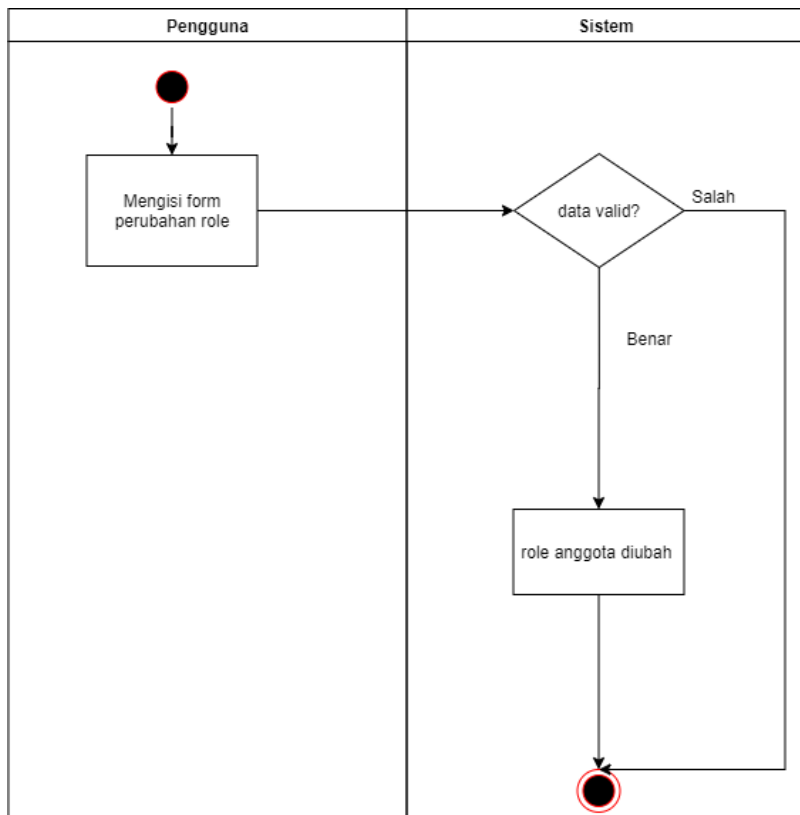
Pengguna dapat menerima undangan dari tim untuk bergabung ke dalam tim pengembang aplikasi. Gambar 4.15 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja penerimaan undangan keanggotaan tim.



Gambar 4.15 F-012 Menerima Undangan Tim

4.4.13 F-013 Mengubah Role Anggota Tim

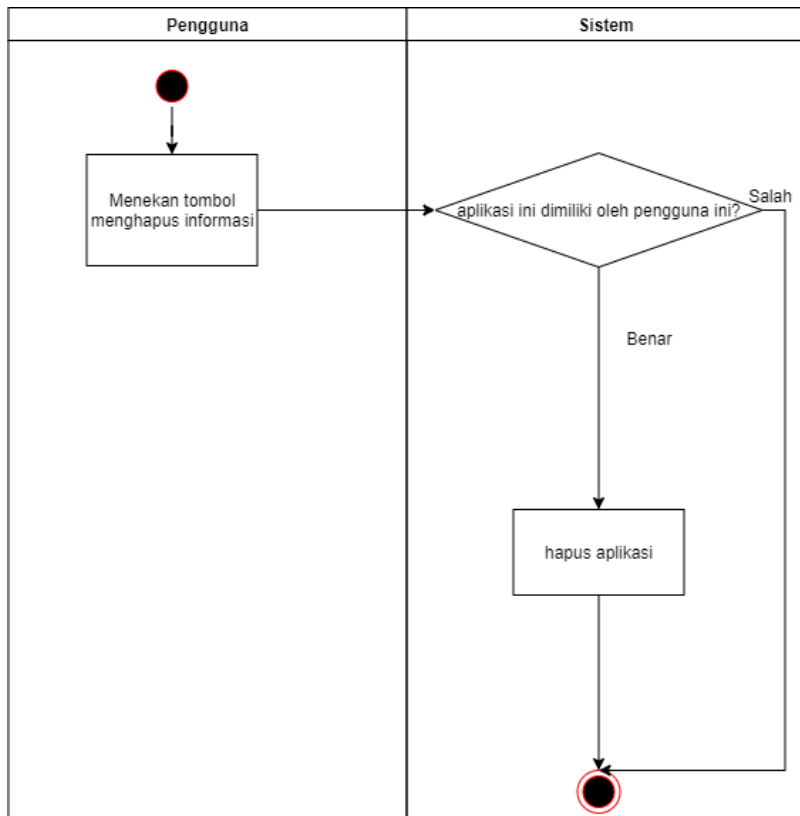
Pengguna dapat mengubah *role* dari anggota tim lainnya. Gambar 4.16 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja perubahan data keanggotaan anggota tim.



Gambar 4.16 F-013 Mengubah Role Anggota Tim

4.4.14 F-014 Menghapus Aplikasi yang Dikelola

Pemilik aplikasi dapat menghapus aplikasi pihak ketiga yang dikembangkan. Gambar 4.17 merupakan diagram aktifitas antara pengguna dan sistem yang menunjukkan alur kerja penghapusan aplikasi yang dikelola.



Gambar 4.17 F-014 Menghapus Aplikasi yang Dikelola

BAB V

IMPLEMENTASI

Bab ini membahas tentang implementasi dari sistem *Storelogy Partner*. Implementasi akan dibagi ke dalam beberapa bagian, yaitu implementasi lapisan model, implementasi lapisan middleware, implementasi lapisan kontrol, dan implementasi antarmuka pengguna.

5.1 Implementasi Model

5.1.1 Lapisan Model *Store*

Lapisan model store ini menyangkut tabel *Store*. Model ini akan menyimpan informasi toko apa saja yang ada di dalam basis data.

```
class Store extends Model
{
    //
}
```

Kode Sumber 5.1 Lapisan Model Store

5.1.2 Lapisan Model *Store_Owner*

Lapisan model *store owner* menyangkut tabel *Store owner*. Model ini akan menyimpan informasi siapa saja yang memiliki akses ke suatu toko di dalam basis data. Model ini memiliki 2 *relationship*, *store_details* yang menghubungkan model *store*, dan *Owner* dengan model *User*.

```
class Store_owner extends Model
{
    public function store_details()
    {
        return $this->belongsTo('App\Store' , 'store_id');
    }
    public function owner()
```

```

    {
        return $this->belongsTo('App\User', 'user_id');
    }
}

```

Kode Sumber 5.2 Lapisan Model Store_Owner

5.1.3 Lapisan Model User

Lapisan model User menyimpan informasi *business name*, *e-mail* dan *password* pengguna. Lapisan ini akan memiliki *relationship User_data* yang menghubungkan model dengan tabel *User_datas*.

```

<?php
class User extends Authenticatable
{
    use Notifiable;
    protected $fillable = [
        'company_name', 'email', 'password',
    ];
    protected $hidden = [
        'password', 'remember_token',
    ];
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
    public function user_data()
    {
        return $this->hasOne('App\User_data');
    }
}

```

Kode Sumber 5.3 Lapisan Model User

5.1.4 Lapisan Model User_data

Lapisan model *User_data* menyimpan informasi pribadi seperti nama lengkap, nomor hp, *website*, dan Alamat pengguna.

```

class User_data extends Model

```



```
{
    protected $primaryKey = 'user_id';
    protected $fillable = ['user_id', 'first_name', 'last_name'
];
}
```

Kode Sumber 5.4 Lapisan Model User_data

5.1.5 Lapisan Model *App_info*

Lapisan model *App_info* menyimpan informasi detail tentang aplikasi pihak ketiga yang terdaftar. Model ini memiliki *Relationship one-to-one* dengan tabel *oauth* dan memiliki *Relationship one-to-one* dengan tabel *App_status*.

```
class App_info extends Model
{
    use HasApiTokens;
    //
    public function oauth()
    {
        return $this-
>belongsTo('App\Oauth_client','oauth_client');
    }
    public function app_status()
    {
        return $this->belongsTo('App\App_status','status');
    }
}
```

Kode Sumber 5.5 Lapisan Model App_info

5.1.6 Lapisan Model *App_status*

Lapisan model *App_status* menyimpan jenis-jenis status aplikasi apa saja yang ada, status ini ditetapkan oleh pengembang, dan memiliki deskripsi dan id.

```
class App_status extends Model
```

```
{
    //
}
```

Kode Sumber 5.6 Lapisan Model App_status

5.1.7 Lapisan Model *Category*

Lapisan model *Category* menyimpan jenis-jenis kategori aplikasi apa saja yang ada, kategori ini ditetapkan oleh pengembang, dan memiliki nama kategori dan id.

```
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{
}
```

Kode Sumber 5.7 Lapisan Model Category

5.1.8 Lapisan Model *App_category*

Lapisan model *App_category* menyimpan kategori aplikasi tersebut.

```
class App_category extends Model
{
    //
    protected $table = 'app_category';
}
```

Kode Sumber 5.8 Lapisan Model App_Category

5.1.9 Lapisan Model *App_store*

Lapisan model *App_store* menyimpan data dan informasi aplikasi yang masuk ke dalam subsistem *Storelogy_AppStore*. Model ini memiliki *Relationship* yang menghubungkan model dengan tabel *App_info*. Model ini memiliki nilai *Primary Key* *app_id*

yang berupa id aplikasi dari *app_info*, hal ini dilakukan untuk mempermudah pencarian *store* yang terdaftar di tabel *App_store*.

```
class App_store extends Model
{
    //
    protected $primaryKey = 'app_id';

    public function app()
    {
        return $this->belongsTo('App\App_info','app_id');
    }
}
```

Kode Sumber 5.9 Lapisan Model *App_store*

5.1.10 Lapisan Model *Partners_team*

Lapisan model *Partners_team* menyimpan semua anggota tim pengembang, aplikasi yang dikembangkan, serta *role*-nya. Lapisan ini memiliki *relationship user* dan *role*.

```
class Partners_team extends Model
{
    //
    public function user()
    {
        return $this->belongsTo('App\User','user_id');
    }
    public function role()
    {
        return $this->belongsTo('App\Role','role_id');
    }
}
```

Kode Sumber 5.10 Lapisan Model *Partners_team*

5.1.11 Lapisan Model *Role*

Lapisan model *Role* menyimpan jenis-jenis *Role* untuk anggota tim apa saja yang ada, *Role* ini ditetapkan oleh pengembang, dan memiliki nama kategori dan id.

```
class Role extends Model
{
    //
}
```

Kode Sumber 5.11 Lapisan Model *Role*

5.1.12 Lapisan Model *Invite*

Lapisan model *Invites* menyimpan semua undangan anggota tim yang dibuat oleh pengembang tim, model ini memiliki *primary key Token* dan memiliki relationship dengan tabel *Role*. Undangan yang dikirimkan akan berisi token undangan dan *role* yang diberikan.

```
class Invite extends Model
{
    protected $primaryKey = 'token';

    protected $fillable = [
        'email', 'token', 'role'
    ];

    public function roles()
    {
        return $this->belongsTo('App\Role', 'role');
    }
}
```

Kode Sumber 5.12 Lapisan Model *Invite*

5.1.13 Lapisan Model *Oauth_Client*

Lapisan model *Oauth_Client* menyimpan data dan informasi nilai kredensial aplikasi terhadap *Oauth Partners Storelogy*. Informasi ini dibutuhkan aplikasi pihak ketiga untuk melakukan transaksi basis data ke basis data *Storelogy*.

```
class Oauth_client extends Model
{
}
```

Kode Sumber 5.13 Lapisan Model Oauth_client

5.2 Implementasi Middleware

5.2.1 CheckAppDeveloper

Lapisan middleware ini akan memeriksa setiap *request*, apakah pengguna yang me-*request* adalah pengembang dari aplikasi tersebut.

```
class CheckAppDeveloper
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        $App_info = App_info::find($request
            ->route('application_id'));
        $Partner_Team = Partners_team::where([
            ['user_id', $request->user()->id],
            ['app_team_id', $App_info->id]
        ]->first();
        if($Partner_Team === null)
```

```

    {
        return redirect('/applications');
    }
    return $next($request);
}
}

```

Kode Sumber 5.14 Lapisan Middleware CheckAppDeveloper

5.2.2 CheckMembershipTeam

Lapisan middleware ini akan memeriksa tiap *request*, apakah pengguna yang me-*request* adalah anggota dari tim pengembang tersebut.

```

class CheckMembershipTeam
{
    public function handle($request, Closure $next)
    {
        $Partner_Team = Partners_team::where([
            ['user_id',$request->user()->id],
            ['app_team_id',$request->route('app_team_id')]
        ]->first();
        if($Partner_Team === null)
        {
            return redirect('/applications');
        }
        return $next($request);
    }
}

```

Kode Sumber 5.15 Lapisan Model CheckMembershipTeam

5.3 Implementasi Lapisan Kontrol

5.3.1 Store Controller

Kontroler ini berfungsi sebagai penggerak logika yang berurusan dengan tampilan *store*. Terdapat fungsi index yang

menampilkan halaman *store* beserta informasi semua toko yang dikelola oleh pengguna.

5.3.1.1 Index

```
public function index(Request $request)
{
    $stores = Store_owner::where('user_id',$request
        ->user()->id)->get();
    return view('pages.stores.stores',['stores'=>
        $stores]);
}
```

Kode Sumber 5.16 Fungsi Index di Store Controller

Fungsi index akan mengambil informasi semua toko yang dikelola pengguna.

5.3.2 New Store Controller

Kontroler ini berfungsi sebagai penggerak logika yang berurusan dengan tampilan *New Store*. Terdapat fungsi index yang menampilkan halaman *new store* dan *create store* untuk membuat toko baru.

5.3.2.1 Index

```
public function index()
{
    return view('pages.stores.new_store');
}
```

Kode Sumber 5.17 Fungsi Index di New Store Controller

Fungsi index akan menampilkan halaman *new_store*.

5.3.2.2 Create_Store

```
public function create_store(Request $request)
{
    $request->validate([
        'store_name'=>['required','max:255','min:3'],
        'login_email'=>['required','email'],
    ]
}
```

```

        'password' => ['required', 'string', 'min:8', 'required_with:password_confirmation', 'confirmed'],
        'company'=>['nullable', 'max:255', 'min:3'],
        'address1'=>['required', 'max:255', 'min:3'],
        'address2'=>['nullable', 'max:255', 'min:3'],
        'city'=>['required', 'max:255', 'min:3'],
        'provence'=>['required', 'max:255', 'min:3'],
        'zip_code'=>['required', 'max:255', 'min:3'],
        'country'=>['required', 'max:255', 'min:3'],
        'phone_number'=>['required'],
        'store_category'=>['required'],
        'found_storelogy'=>['required'],
        'facebook_url'=>['nullable'],
        'instagram_url'=>['nullable'],
    ]));
    $address = "{$request->company} {$request->address1} {$request->address2} {$request->city}, {$request->provence}, {$request->country}, {$request->zip_code}";

    $response = Http::post('https://master.dev.storelogy.com/api/create_store',[
        'first_name'=>\Auth::user()->user_data->first_name,
        'last_name'=>\Auth::user()->user_data->last_name,
        'email'=>$request->login_email,
        'password'=>$request->password,
        'store_name'=>$request->store_name,
        'phone_number'=>$request->phone_number,
        'store_category'=>$request->store_category,
        'country'=>$request->country,
        'address'=> $address,
        'found_storelogy'=>$request->found_storelogy,
        'facebook_url'=>$request->facebook_url,
        'instagram_url'=>$request->instagram_url,
        'ios_app'=>'0',
        'android_app'=>'0',
    ]);

```



```

        'remove_branding'=>'0',
        'package_id'=>'3',
    ]);

    //create new Store
    $store = new Store();
    $store->name = $request->store_name;
    $store->link = $response['url'];
    $store->save();

    //create new store owner
    $store_owner = new Store_owner();
    $store_owner->user_id = \Auth::user()->id;
    $store_owner->store_id = $store->id;
    $store_owner->save();
    return redirect('/stores/');
}

```

Kode Sumber 5.18 Fungsi Create Store di new

Create_store akan membuat toko baru dengan mengirimkan *request* ke *Storelogy API* yang akan memberikan *url* toko yang dibuat serta mendaftarkan pengguna sebagai salah satu pemilik toko.

5.3.3 Application Controller

Kontroler ini berfungsi sebagai penggerak logika yang berurusan dengan tampilan *Application*. Terdapat fungsi *index* yang menampilkan halaman *Application* yang menunjukkan aplikasi apa saja yang dikembangkan pengguna.

5.3.3.1 Index

```

public function index(Request $request)
{
    $applications = App_info::where('creator_id',$request->user()->id)->get();
    return view('pages.applications.applications',['applications'=>$applications]);
}

```

Kode Sumber 5.19 Fungsi Index di Application Controller

Fungsi index akan mengambil informasi semua aplikasi pihak ketiga yang dikelola pengguna. Data didapat menggunakan model *App_info*.

5.3.4 New Application Controller

Kontroler ini berfungsi sebagai penggerak logika yang berurusan dengan tampilan *New Application*. Terdapat fungsi index yang menampilkan halaman *New Application* serta *create* untuk membuat data aplikasi baru.

5.3.4.1 Index

```
public function index()
{
    $categories = Category::All();
    return view('pages.applications.new_application', ['categories' => $categories]);
}
```

Kode Sumber 5.20 Fungsi Index di New Application Controller

Fungsi index akan mengambil informasi semua kategori menggunakan model *Category* dan menampilkan halaman *new_application* beserta data kategori.

5.3.4.2 Create Application

```
public function create(Request $request)
{
    $request->validate([
        'app_name' => ['required', 'max:255'],
        'type' => ['required'],
        'category' => ['required'],
        'app_url' => ['required', 'max:255', 'url'],
        'whitelisted_redirection_url' => ['required', 'max:255', 'url'],
    ]);
}
```

```

        'oauth_client' => ['required'],
    ]));

    //create app
    $app = new App_info();
    $app->name = $request->app_name;
    $app->type = $request->type;
    $app->api_contact_email = \Auth::user()->email;
    $app->url = $request->app_url;
    $app->oauth_client = $request->oauth_client;
    $app->status = "1";
    $app->creator_id = \Auth::user()->id;
    $app->slug = Str::slug($request->app_name);
    $app->save();

    //create App Category
    $app_category = new App_category();
    $app_category->app_id = $app->id;
    $app_category->category_id = $request->category;
    $app_category->save();

    //create new team membership
    $Partner_team = new Partners_team();
    $Partner_team->user_id = \Auth::user()->id;
    $Partner_team->role_id = 1; //owner
    $Partner_team->app_team_id = $app->id;
    $Partner_team->save();

    return true;
}

```

Kode Sumber 5.21 Fungsi Create di Store Controller

Create akan membuat data aplikasi pihak ketiga baru di basis data, kemudian kunci *oauth* akan didapat dari tampilan *create app* yang melakukan API *request* dengan *Server OAUTH Storelogy*.

5.3.5 Detailed Application Controller

Kontroler ini berfungsi sebagai penggerak logika yang berurusan dengan tampilan *Detailed Application*, yang akan menunjukkan menu tambahan untuk aplikasi yang dipilih. Terdapat fungsi index yang menampilkan halaman *Detailed Application*.

5.3.5.1 Index

```
public function index(Request $request, $application_id)
{
    $data['application'] = App_info::find($application_id);
    $data['team_members'] = Partners_team::where('app_team_id', $data['application']->id)->get();
    $data['roles'] = Role::All();
    $data['invites'] = Invite::where('app_team_id', $application_id)->get();
    return view('pages.applications.detailed_application', ['data'=>$data]);
}
```

Kode Sumber 5.22 Fungsi Index di Detailed Application Controller

Detailed Application mengambil informasi-informasi berupa anggota yang diundang dan anggota tim. Anggota tim didapat dari model *Partners_team*, sedangkan undangan didapat dari model *Invites*.

5.3.6 Edit Application Controller

Kontroler ini berfungsi sebagai penggerak logika yang berurusan dengan tampilan *Edit Application*, yang akan menunjukkan form untuk mengubah detail aplikasi yang dipilih. Terdapat fungsi index yang menampilkan halaman *Edit Application*, edit untuk mengubah informasi aplikasi, *Edit_app_icon* untuk mengubah icon, dan *delete* untuk menghapus aplikasi.

5.3.6.1 Index

```
public function index(Request $request, $application_id)
{
    $application = App_info::find($application_id);
    return view('pages.applications.edit_application', [
        'application'=>$application]);
}
```

Kode Sumber 5.23 Fungsi Index di Edit Application Controller

Edit Application membutuhkan informasi-informasi tentang aplikasi yang dipilih, melalui models *App_info*. Fungsi ini akan menampilkan halaman *edit_application*.

5.3.6.2 Edit

```
public function edit(Request $request, $application_id)
{
    $request->validate([
        'app_name' => ['required', 'max:255'],
        'api_contact_email' => ['nullable', 'max:255', 'email'],
        'app_url' => ['required', 'max:255', 'url'],
        'prefrence_url' => ['nullable', 'max:255', 'url'],
        'whitelisted_redirection_url' => ['required', 'max:255', 'url'],
    ]);
    $app = App_info::find($application_id);
    $app->name = $request->app_name;
    $app->api_contact_email = $request->api_contact_email;
    $app->prefrence_url = $request->prefrence_url;
    $app->url = $request->app_url;
    $app->oauth->redirect = $request->whitelisted_redirection_url;
    $app->oauth->save();
    $app->save();
    return redirect('/applications/'.$app->id.'/edit');
}
```

Kode Sumber 5.24 Fungsi Edit di Edit Application Controller

Edit akan merubah informasi dari aplikasi yang dipilih, terdapat validasi nilai *request* untuk melihat apakah nilai yang dikirimkan sudah sesuai atau belum.

5.3.6.3 Edit App Icon

```
$file = $request->file('icon');
$upload_directory = 'img/application_icon/'.$application_id;
$file_name=str_replace(' ', '', $file->getClientOriginalName());
$file->move($upload_directory,$file_name);

$app = App_info::find($application_id);
$app->icon = $upload_directory.'/'.$file_name;
$app->save();
return back();
```

Kode Sumber 5.25 Fungsi Edit App Icon di Edit Application Controller

Edit app icon akan mengubah file gambar yang menjadi logo atau ikon dari aplikasi tersebut, file gambar ini akan disimpan di server, sehingga basis data cukup mencatat nama filenya saja.

5.3.6.4 Delete

```
public function delete($application_id)
{
    $app = App_info::find($application_id);
    if($app->creator_id==Auth::user()->id)
    {
        $app->delete();
        return redirect('/applications/');
    }
    else abort(401);
}
```

Kode Sumber 5.26 Fungsi Delete di Edit Application Controller

Delete akan menghapus data aplikasi yang dibuat, data yang dihapus juga akan menghapus *relationship* lainnya berupa *Partners_team*, *Invitation*, dan *App Store*.

5.3.7 Edit Application Listing Controller

Kontroler ini berfungsi sebagai penggerak logika yang berurusan dengan tampilan *Edit Application Listing*, yang akan menunjukkan form untuk mengubah listing aplikasi yang mau ditampilkan di *App Store* seperti gambar penggunaan, deskripsi aplikasi, dan lainnya untuk memudahkan pengguna mengetahui aplikasi tersebut. Terdapat fungsi *index* yang menampilkan halaman *Edit Application Listing*, *submit* untuk mengubah informasi listing aplikasi.

5.3.7.1 Index

```
public function index($app_id)
{
    $appinfo = App_store::find($app_id);
    return view('pages.applications.applisting', ['appinfo' => $appinfo]);
}
```

Kode Sumber 5.27 Fungsi Index di Edit Application Listing Controller

Edit Application Listing membutuhkan informasi-informasi tentang aplikasi yang dipilih, melalui models *App_info*. Fungsi ini akan menampilkan tampilan *AppListing*.

5.3.7.2 Submit

```
public function submit(Request $request)
{
    $this->validate($request,[
        'app_id'=>'required'
    ]);
}
```

```

$appstore = App_store::find($request->app_id);

$logo = $request->file('logo');
if($logo)
{
    File::delete($appstore->logo);
    $logo->move('app_data/'.$request->
>app_id, 'logo.'.$logo->getClientOriginalExtension());
    $appstore->logo = 'app_data/'.$request->
>app_id.'/logo.'.$logo->getClientOriginalExtension();
}

$banner1 = $request->file('banner1');
if($banner1)
{
    File::delete($appstore->banner1);
    $banner1->move('app_data/'.$request->
>app_id, 'banner1.'.$banner1->getClientOriginalExtension());
    $appstore->banner1 = 'app_data/'.$request->
>app_id.'/banner1.'.$banner1->getClientOriginalExtension();
}

$banner2 = $request->file('banner2');
if($banner2)
{
    File::delete($appstore->banner2);
    $banner2->move('app_data/'.$request->
>app_id, 'banner2.'.$banner2->getClientOriginalExtension());
    $appstore->banner2 = 'app_data/'.$request->
>app_id.'/banner2.'.$banner2->getClientOriginalExtension();
}

$banner3 = $request->file('banner3');
if($banner3)
{
    File::delete($appstore->banner3);

```



```

        $banner3->move('app_data/'.$request-
>app_id, 'banner3.'.$banner3->getClientOriginalExtension());
        $appstore->banner3 = 'app_data/'.$request-
>app_id.'/banner3.'.$banner3->getClientOriginalExtension();
    }

    $appstore->description = $request->description;
    $appstore->video_url = $request->video_url;
    $appstore->banner_head_1 = $request->banner_head_1;
    $appstore->banner_head_2 = $request->banner_head_2;
    $appstore->banner_head_3 = $request->banner_head_3;
    $appstore->banner_text_1 = $request->banner_text_1;
    $appstore->banner_text_2 = $request->banner_text_2;
    $appstore->banner_text_3 = $request->banner_text_3;
    $appstore->faq_url = $request->faq_url;
    $appstore->web_url = $request->web_url;
    $appstore->pp_url = $request->pp_url;
    $appstore->support_number = $request-
>support_number;
    $appstore->support_email = $request->support_email;
    $appstore->about_text = $request->about_text;
    $appstore->updated_at = Carbon::now();
    $appstore->save();

    $appinfo = App_info::find($request->app_id);
    $appinfo->name = $request->app_name;
    $appinfo->slug = Str::slug($request->app_name);
    $appinfo->save();

    return back();
}

```

Kode Sumber 5.28 Fungsi Submit di Edit Application Listing Controller

Fungsi Submit akan mengubah informasi dari *app listing*.

5.3.8 Manage Team Controller

Kontroler ini berfungsi sebagai penggerak logika yang menyangkut tampilan *Manage_Team_Controller* yang dapat mengatur anggota tim pengembang, terdapat fungsi index untuk menampilkan halaman, update_team_member untuk mengubah detail anggota tim, dan delete_team_member untuk menghapus anggota tim.

5.3.8.1 Index

```
public function index($app_team_id,$user_id)
{
    $roles = Role::All();

    $member = Partners_team::where([
        ['user_id',$user_id],
        ['app_team_id',$app_team_id],
    ])->first();

    return view('pages.teams.manage_team_member',
        ['member'=>$member],
        ['roles'=>$roles]
    );
}
```

Kode Sumber 5.29 Fungsi Index di Manage Team Controller

Index akan menunjukkan halaman *manage_team* serta mendapatkan informasi anggota-anggota tim beserta *role*-nya.

5.3.8.2 Update Team Member

```
public function update_team_member(Request $request, $app_team_id,$user_id)
{
    $member = Partners_team::where([
        ['user_id',$user_id],
        ['app_team_id',$app_team_id],
```

```

    ]->update(
        ['role_id'=>$request->role]
    );

    return redirect()->back();
}

```

Kode Sumber 5.30 Fungsi Update Team Member di Manage Team Controller

Fungsi ini akan mengubah informasi dari anggota tim di basis data *partners_team*. Se jauh ini, pengguna hanya bisa mengubah *role*.

5.3.8.3 Delete Team Member

```

public function delete_team_member(Request $request, $app_team_id,$user_id)
{
    $member = Partners_team::where([
        ['user_id',$user_id],
        ['app_team_id',$app_team_id],
    ]->delete());
    return redirect('/applications/');
}

```

Kode Sumber 5.31 Fungsi Delete Team Member di Manage Team Controller

Fungsi ini akan menghapus informasi dari *Partners_team* yang akan menghapus keanggotaan seseorang terhadap tim itu.

5.3.9 Team Controller

Kontroler ini berfungsi sebagai penggerak logika untuk penerimaan keanggotaan tim.

5.3.9.1 Accept Invitation

```

public function accept_invitation(Request $request,$token)
{
    $invite = Invite::find($token);
    if($invite){
        //check if exist
        if(Partners_team::where([
            ['user_id','=',\Auth::user()->id],

```

```

        ['app_team_id','=', $invite->app_team_id]
    ]->first()){
        dd("exist");
    }
    else{
        $Partner_team = new Partners_team();
        $Partner_team->user_id = \Auth::user()->id;
        $Partner_team->role_id = $invite->role;
        $Partner_team->app_team_id = $invite-
>app_team_id;
        $Partner_team->save();

        return redirect('/applications/');
    }
}
else
{
    return redirect('/home/');
}
}

```

Kode Sumber 5.32 Fungsi Accept Invitation di Team Controller

Fungsi ini akan memeriksa apakah token dan e-mail pengguna sesuai dengan undangan, apabila benar anggota tim baru akan ditambahkan di basis data.

5.3.10 Detailed Team Controller

Kontroler ini berfungsi sebagai penggerak logika yang berurusan dengan fungsi untuk membuat undangan dan menghapus undangan.

5.3.10.1 Create_Invite

```

public function create_invite(Request $request,$app_tea
m_id)
{
    $token = Str::random(16);
    $invite = new Invite();

```

```

        $invite->email=$request->team_member_email;
        $invite->app_team_id=$app_team_id;
        $invite->role=$request->role;
        $invite->token=$token;
        $invite->save();
        //send email
        $details['email']=$invite->email;
        $details['token']=$token;
        dispatch(new \App\Jobs\SendEmail($details,'invite'))
;
        return redirect()->back();
    }

```

Kode Sumber 5.33 Fungsi Create Invite di Detailed Team Controller

Fungsi ini dipakai untuk membuat undangan ke dalam anggota tim baru.

5.3.10.2 Delete Invite

```

public function delete_invite($app_team_id,$id)
{
    $invite = Invite::where('id',$id);
    $invite->delete();
    return redirect()->back();
}

```

Kode Sumber 5.34 Fungsi Delete Invite di Detailed Team Controller

Fungsi ini dipakai untuk menghapus undangan ke dalam anggota tim.

5.3.11 Settings Controller

Kontroler ini berfungsi sebagai penggerak logika yang dipasangkan ke view *Settings*. Pengguna dapat mengubah informasi pribadi yang terdaftar di basis data.

5.3.11.1 Index

```

public function index()
{
    $data['user'] = Auth::user();
}

```

```

        return view('pages.settings.settings',
            ['data'=>$data]
        );
    }

```

Kode Sumber 5.35 Fungsi Index di Settings Controller

Fungsi index akan mendapatkan informasi-informasi pengguna dan menampilkan halaman *settings*.

5.3.11.2 Update User

```

public function update_user(Request $request)
{
    $request->validate([
        'first_name' => ['required','max:255'],
        'last_name' => ['nullable','max:255'],
        'email' => ['required', 'string', 'email', 'max:
255', 'unique:users'],
    ]);
    $user = Auth::user();
    $user_data = $user->user_data;
    $user_data->first_name = $request->
>first_name;
    $user_data->last_name = $request->
>last_name;
    $user->email = $request->email;
    $user_data->save();
    $user->save();

    //if email change, do verification!
    return Redirect('settings');
}

```

Kode Sumber 5.36 Fungsi Update User di Settings Controller

Fungsi ini akan mengubah informasi pribadi pengguna, dibutuhkan model *user* untuk mengubah nilainya. Diberikan validasi nilai untuk memeriksa apakah nilai yang dikirimkan sesuai atau tidak.

5.3.11.3 Update Password

```
public function update_password(Request $request)
{
    $request->validate([
        'old_password' => ['required', 'string'],
        'password' => ['required', 'string', 'min:8', 'required_with:password_confirmation', 'confirmed'],
    ]);
    $user = Auth::user();

    if(Hash::check($request->old_password, $user->password)) {

        return redirect('/settings');
    } else {
        return redirect()->back()->withErrors(['password' => 'Old password is not correct']);
    }

    $user_data->save();
    $user->save();

    //if email change, do verification!
    return Redirect('settings');
}
```

Kode Sumber 5.37 Fungsi Update Password di Settings Controller

Fungsi ini akan mengubah *password* pengguna. Dibutuhkan model *user* untuk mengubah nilainya. Diberikan validasi nilai untuk memeriksa apakah nilai yang dikirimkan dan *password* lama sesuai, serta apakah password baru dan konfirmasi password sama.

5.3.11.4 Update User Contact

```
public function update_user_contact(Request $request)
{
    $request->validate([
        'website' => ['nullable','max:255'],
        'phone_number' => ['required','string','max:255'],
        'company_name' => ['required','string','max:255'],
    ]);

    $User_contact = Auth::user()->user_data;
    $User_contact->website = $request->website;
    $User_contact->phone_number = $request->phone_number;
    $User_contact->company_name = $request->company_name;
    $User_contact->save();
    return Redirect('settings');
}
```

Kode Sumber 5.38 Fungsi Update User Contact di Settings Controller

Fungsi ini akan mengubah informasi kontak pengguna. Dibutuhkan model *user* untuk mengubah nilainya. Diberikan validasi nilai untuk memeriksa apakah nilai yang dikirimkan sesuai atau tidak.

5.3.11.5 Update User Address

```
public function update_user_address(Request $request)
{
    $request->validate([
        'address1' => ['required','string','max:255'],
        'provence' => ['required','string','max:255'],
        'country' => ['required','string','max:255'],
        'address2' => ['nullable','string','max:255'],
    ]);
}
```



```

        'city' => ['required','string','max:255'],
        'zip_code' => ['required','string','max:255'],
    ));
    $User_contact = Auth::user()-
>user_data;
    $User_contact->address1 = $request-
>address1;
    $User_contact->address2 = $request-
>address2;
    $User_contact->city = $request->city;
    $User_contact->provence = $request-
>provence;
    $User_contact->country = $request->country;
    $User_contact->zip_code = $request-
>zip_code;
    $User_contact->save();
    return Redirect('settings');
}

```

Kode Sumber 5.39 Fungsi Update User Address di Settings Controller

Fungsi ini akan mengubah informasi alamat pengguna, dibutuhkan model user untuk mengubah nilainya. Diberikan validasi nilai untuk memeriksa apakah nilai yang dikirimkan sesuai atau tidak.

5.3.11.6

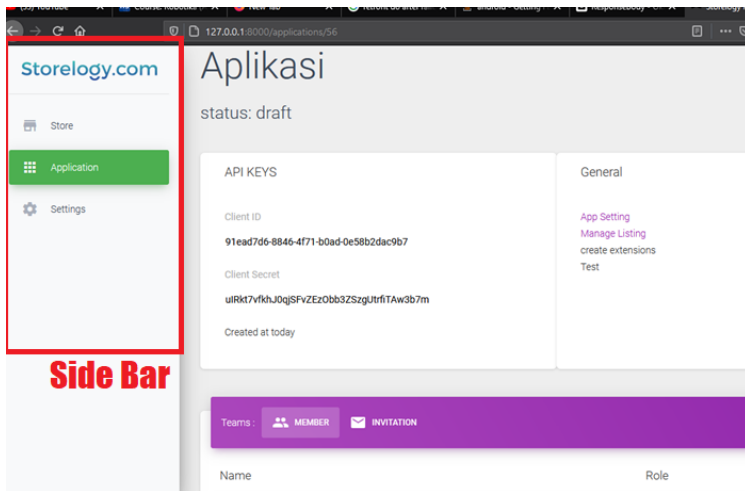
5.4 Implementasi Antar Muka Pengguna

Implementasi antar muka pengguna akan menggunakan bootstrap dan template *material dashboard*. Implementasi Antar muka dibagi dua yaitu *Layout* yang memiliki alat navigasi bantuan dari halaman seperti *navbar* dan *sidebar*, serta halaman itu sendiri yang berisi konten.

5.4.1 Layout

5.4.1.1 Side Bar

Sidebar adalah navigasi yang terletak di kiri halaman apabila pengguna sudah *login* ditunjukkan pada Gambar 5.1



Gambar 5.1 Side bar

5.4.1.2 Nav Bar

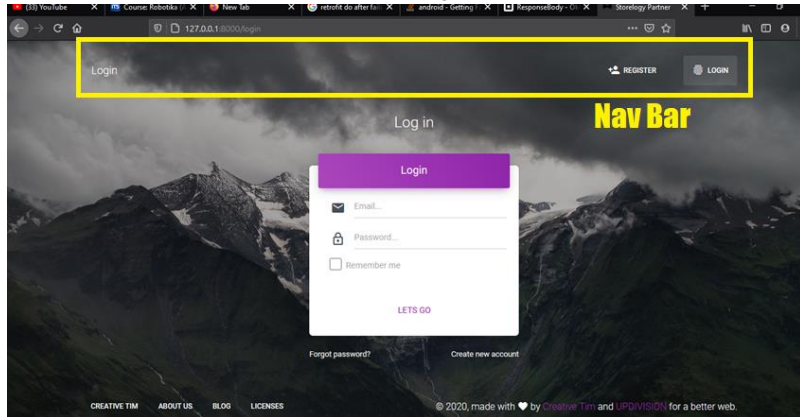
Navbar adalah navigasi yang terletak di atas halaman. Navbar terbagi dua berdasarkan pengguna sudah *login*, pada Gambar 5.2, atau belum login, pada Gambar 5.3.

Sudah Login



Gambar 5.2 Navbar ketika sudah login

Belum Login



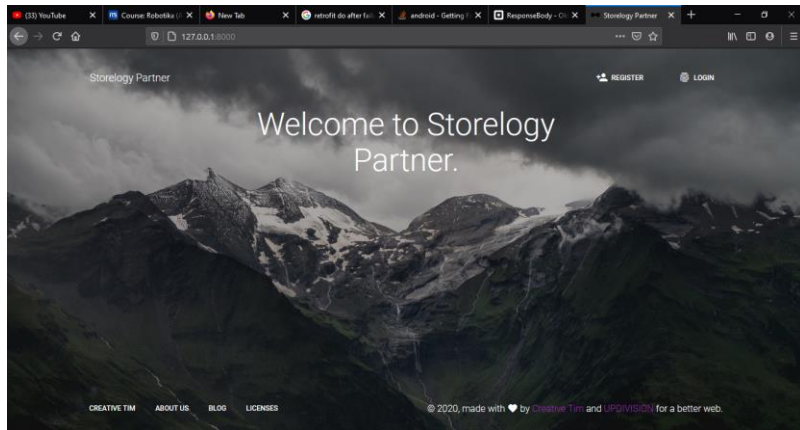
Gambar 5.3 Navbar Ketika Belum Login

5.4.2 Halaman

Halaman berisikan konten dari *website* berdasarkan rute sekarang. Halaman akan merangkap layout dan menampilkan navbar sesuai apakah pengguna sudah *login* atau belum.

Welcomes

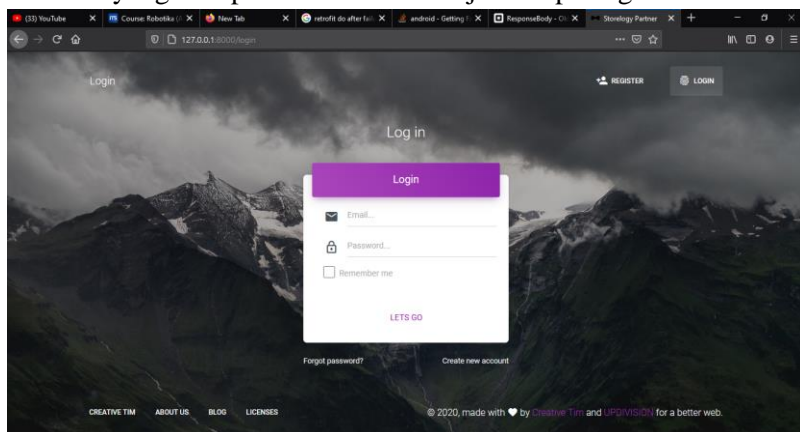
Halaman welcome menunjukkan tampilan pertama bagi user yang mengakses *Storelogy Partners*. Terdapat di rute '/'. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.4



Gambar 5.4 Halaman Welcomes

Login

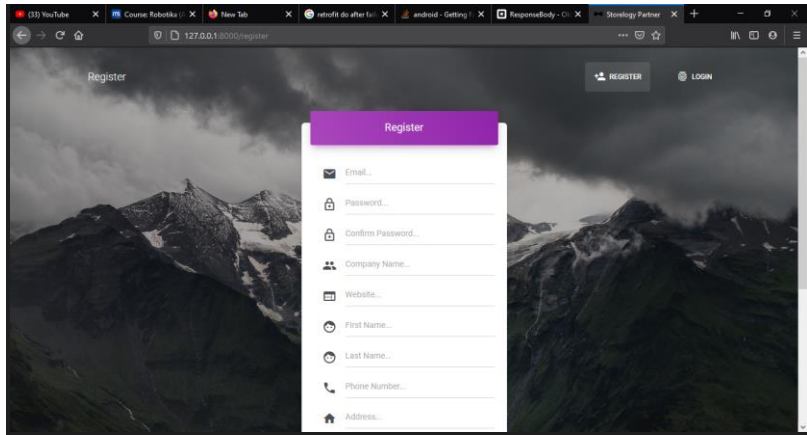
Tampilan login akan menunjukkan form untuk login. Halaman ini dapat diakses melalui rute '/login'. Tampilan dan formulir yang diimplementasikan ditunjukkan pada gambar 5.5.



Gambar 5.5 Halaman Login

Register

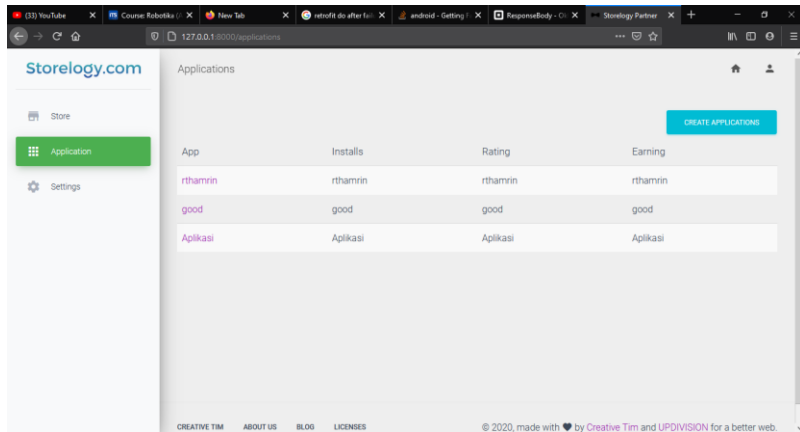
Tampilan register akan menunjukkan form untuk mendaftarkan akun. Halaman ini dapat diakses melalui rute ‘/register’. Tampilan register ditunjukkan pada Gambar 5.6.



Gambar 5.6 Halaman Register

Application

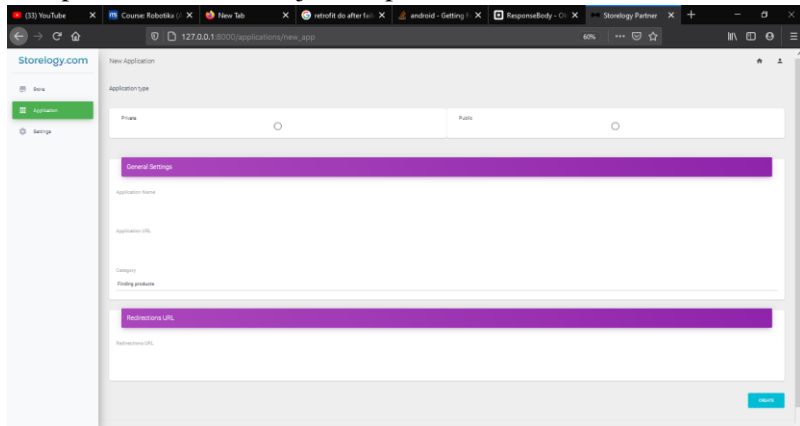
Tampilan *application* akan menunjukkan daftar aplikasi apa saja yang dikelola pengguna. Halaman ini dapat diakses melalui rute ‘/applications’. Tampilan applications ditunjukkan pada Gambar 5.7.



Gambar 5.7 Halaman Applications

New Application

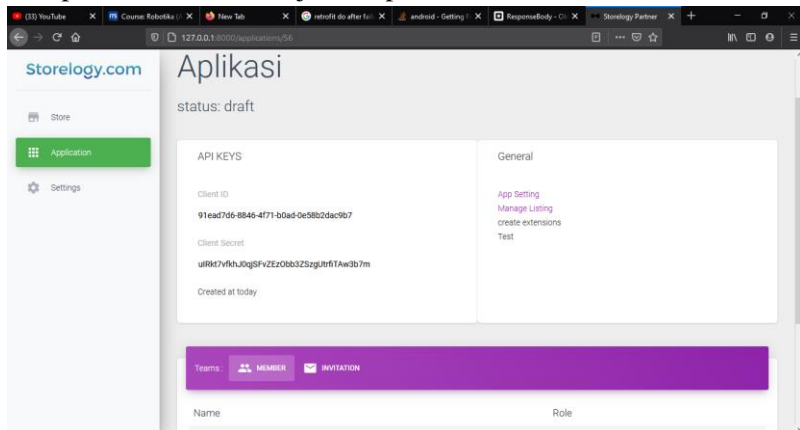
Tampilan *new application* akan menunjukkan form untuk mendaftarkan aplikasi pihak ketiga yang baru. Halaman ini dapat diakses melalui rute `‘/applications/new_app’`. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.8.



Gambar 5.8 Halaman New Applications

Detailed Application

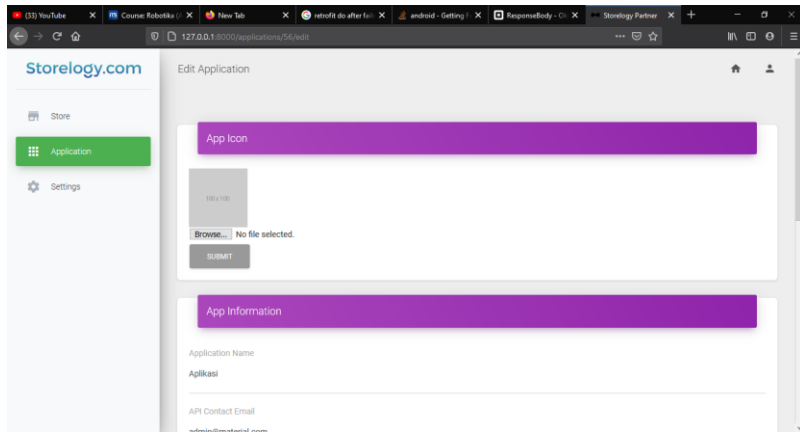
Tampilan *Detailed application* akan menunjukkan informasi-informasi aplikasi pihak ketiga beserta menu-menu lainnya. Apabila aplikasi tersebut merupakan aplikasi publik, menu *manage listing* akan ditampilkan. Terdapat juga *dropdown* untuk mengundang anggota tim baru. Halaman ini dapat diakses melalui rute `‘/applications/{id_aplikasi}’`. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.9



Gambar 5.9 Halaman Detailed Applications

Edit Application

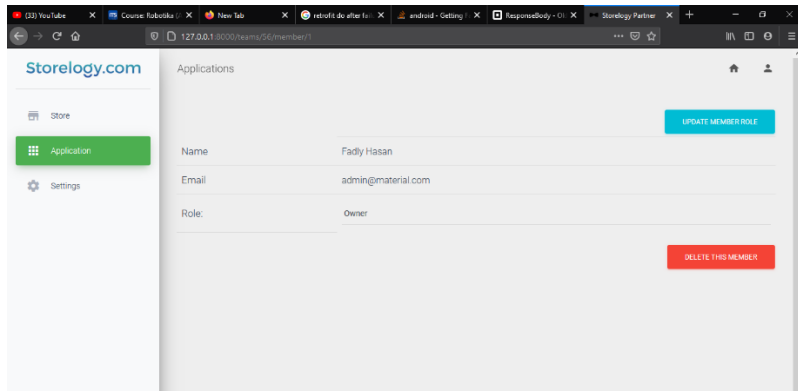
Tampilan *Edit Application* akan menunjukkan formulir untuk mengubah informasi aplikasi. Halaman ini dapat diakses melalui rute `‘/applications/{id Aplikasi}/edit’`. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.10



Gambar 5.10 Halaman Edit Applications

Manage Team Member

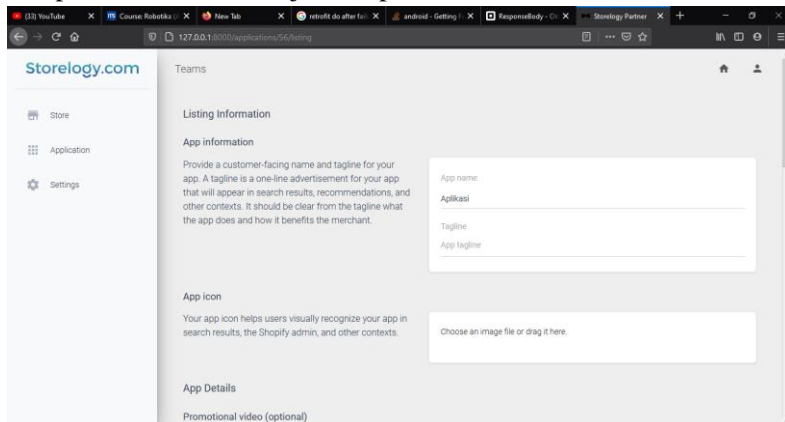
Manage Team Member menunjukkan detail satu anggota tim dan rolenya, pengguna dapat mengubah status dari anggota tim tersebut. Halaman ini dapat diakses melalui rute `‘/teams/{id Aplikasi}/member/{id_member}’`. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.11.



Gambar 5.11 Halaman Manage Team Member

App Listing

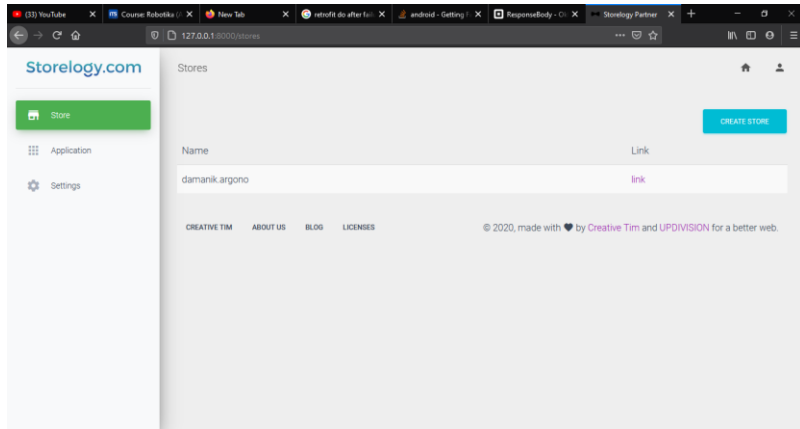
Tampilan *App Listing* akan menunjukkan form untuk mengubah listing aplikasi. Halaman ini dapat diakses melalui rute ‘`/applications/{id Aplikasi}/listing`’. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.12.



Gambar 5.12 Halaman AppListing

Store

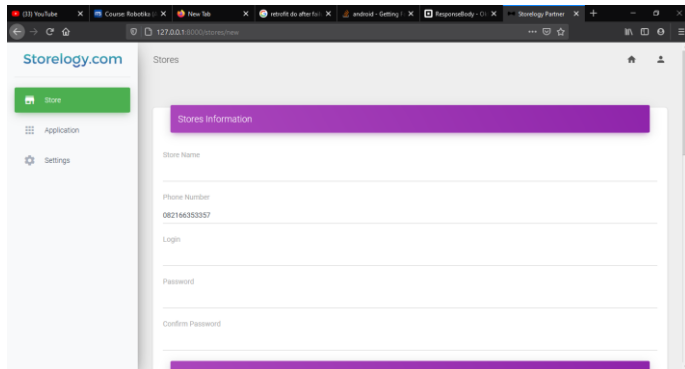
Tampilan store akan menunjukkan daftar toko apa saja yang dikelola pengguna. Halaman ini dapat diakses melalui rute `‘/store’`. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.13.



Gambar 5.13 Halaman Store

New Store

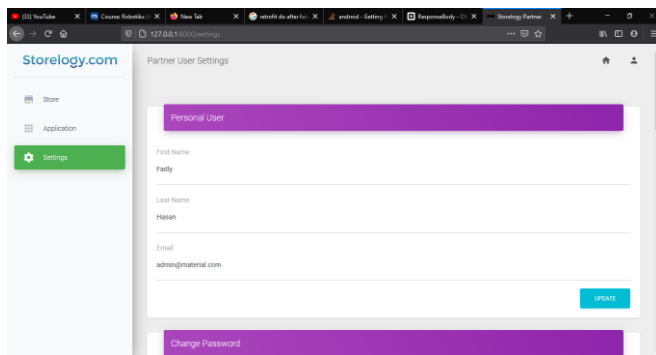
Tampilan *New Store* akan menunjukkan formulir untuk membuat toko baru. Halaman ini dapat diakses melalui rute `‘/stores/new’`. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.14.



Gambar 5.14 Halaman New Store

Settings

Tampilan *Settings* akan menunjukkan form untuk mengubah informasi akun. Halaman ini dapat diakses melalui rute `‘/settings’`. Tampilan yang diimplementasikan ditunjukkan pada Gambar 5.15



Gambar 5.15 Halaman Settings

[Halaman ini sengaja dikosongkan]

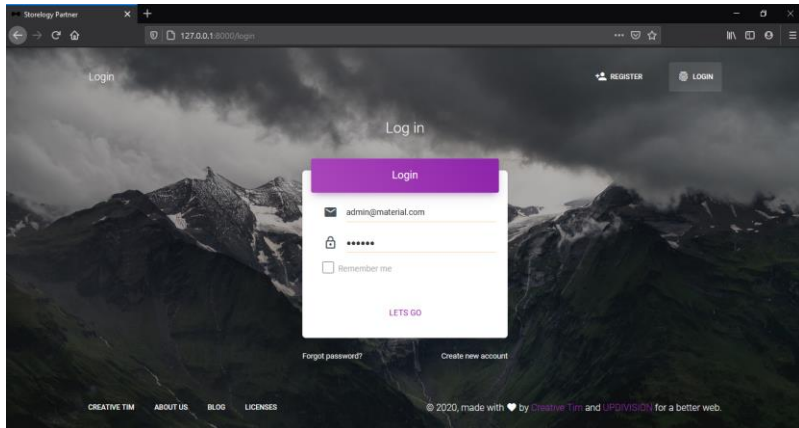
BAB VI

PENGUJIAN DAN EVALUASI

Bab ini memaparkan proses pengujian *Storelogy Partner*. Pengujian dilakukan untuk memastikan kebenaran dan kesesuaian hasil eksekusi perangkat lunak berdasarkan semua kebutuhan fungsionalnya. Pengujian dilakukan pada tanggal 4 November 2020 Pukul 13.00 WIB, Akun yang akan digunakan ada 3, masing masing memiliki e-mail yang berbeda-beda.

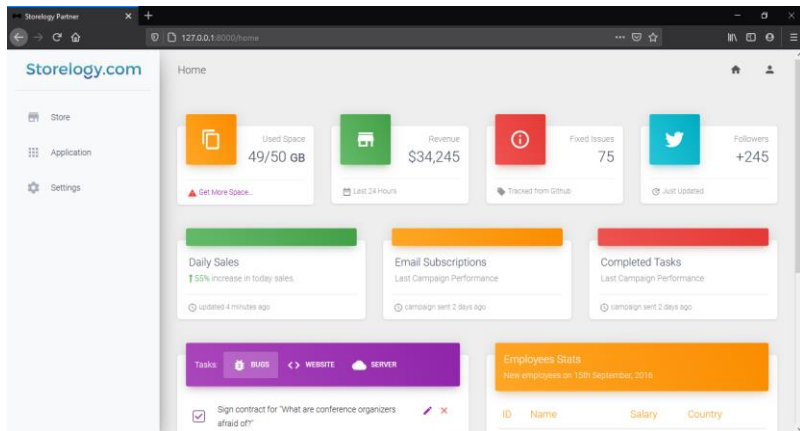
6.1 F-001 Login Partner

Pengguna *login* menggunakan email admin@material.com dengan *password* ‘rahasia’ seperti yang ditunjukkan pada gambar 6.1. Pengujian terpenuhi apabila pengguna berhasil *login* dan diarahkan ke halaman *home*.



Gambar 6.1 Pengujian Login Partner

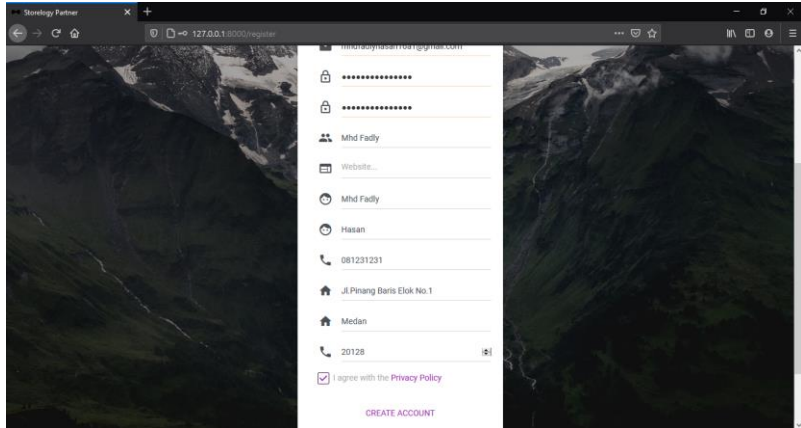
Hasil pengujian ditunjukkan pada Gambar 6.2. Terlihat bahwa pengguna diarahkan ke halaman 'home', dimana tampilan *home* hanya dapat diakses oleh pengguna yang sudah *Login*. Sehingga dapat disimpulkan bahwa pengujian terpenuhi.



Gambar 6.2 Hasil Pengujian Login Partner

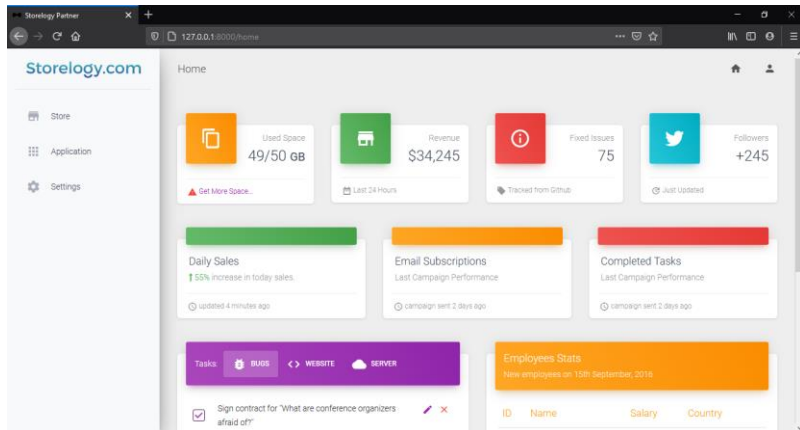
6.2 F-002 Membuat Akun

Pengguna mendaftar menggunakan e-mail (mhdfadlyhasan16a1@gmail.com) dan data pribadi penulis seperti yang ditunjukkan pada gambar 6.3. Pengujian terpenuhi apabila pengguna berhasil registrasi dan diarahkan ke halaman *home*.

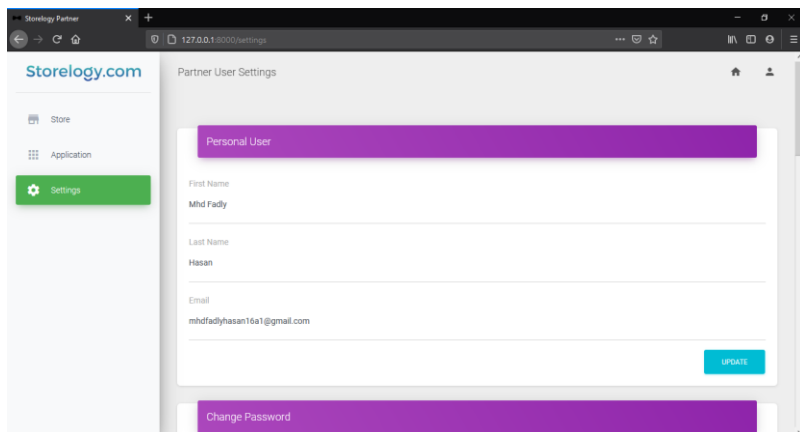


Gambar 6.3 Pengujian Membuat Akun

Hasil pengujian terpenuhi, ditunjukkan pada Gambar 6.4, terlihat bahwa pengguna diarahkan ke halaman '*home*', serta pada gambar 6.5 dimana informasi pengguna sekarang adalah pengguna yang memiliki informasi pribadi penulis



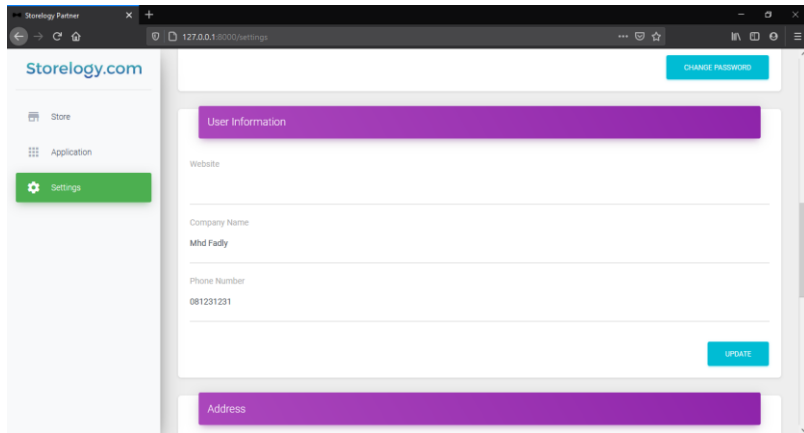
Gambar 6.4 Hasil Pengujian Membuat Akun



Gambar 6.5 Hasil Pengujian membuat Akun beserta detail akun

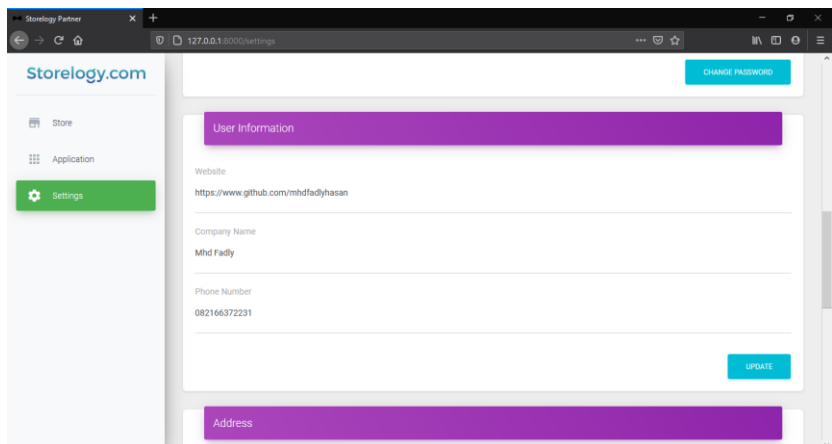
6.3 F-003 Mengubah data diri

Pengguna mengubah informasi akun pribadi penulis yang akan mengubah informasi nomor hp, website dan nama bisnis seperti yang ditunjukkan pada gambar 6.6. Pengujian terpenuhi apabila informasi nomor hp, website, dan nama bisnis berhasil diubah.



Gambar 6.6 Pengujian Mengubah Data Diri

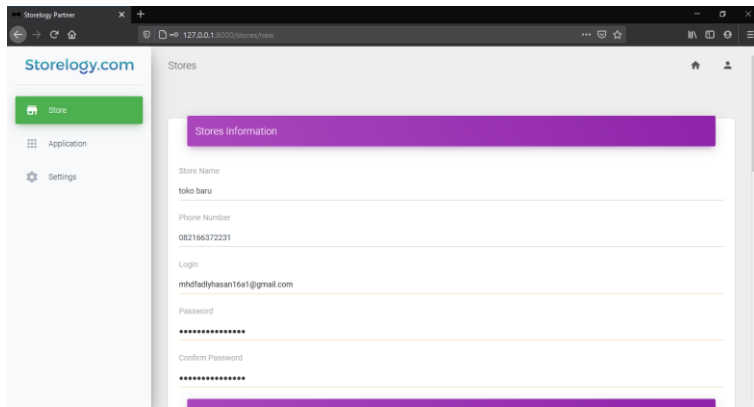
Hasil pengujian terpenuhi, ditunjukkan pada Gambar 6.7, terlihat bahwa informasi nomor hp, nama bisnsi, dan website berhasil diubah.



Gambar 6.7 Hasil Pengujian Mengubah Data Diri

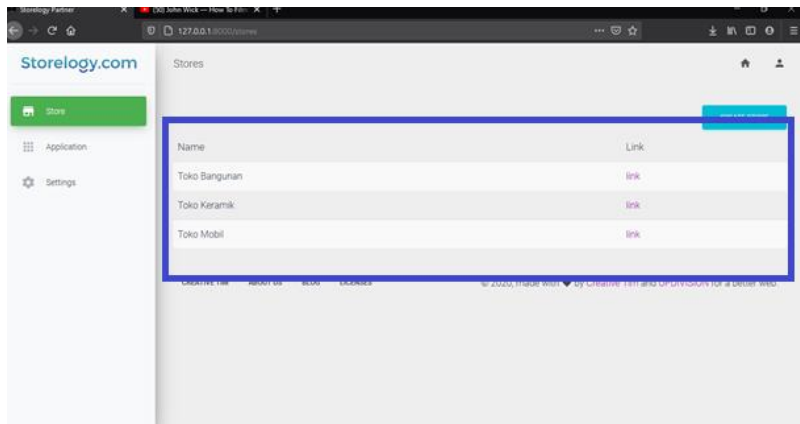
6.4 F-004 Membuat toko

Pengguna membuat tiga toko baru, bernama ‘toko bangunan’, ‘toko keramik, dan ‘toko mobil’ berkategori bebas, menggunakan akun dengan email mhdfadlyhasan16a1@gmail.com seperti yang ditunjukkan pada gambar 6.8. Pengujian terpenuhi apabila toko baru telah terdaftar di tampilan pengguna.



Gambar 6.8 Pengujian Membuat Toko Baru

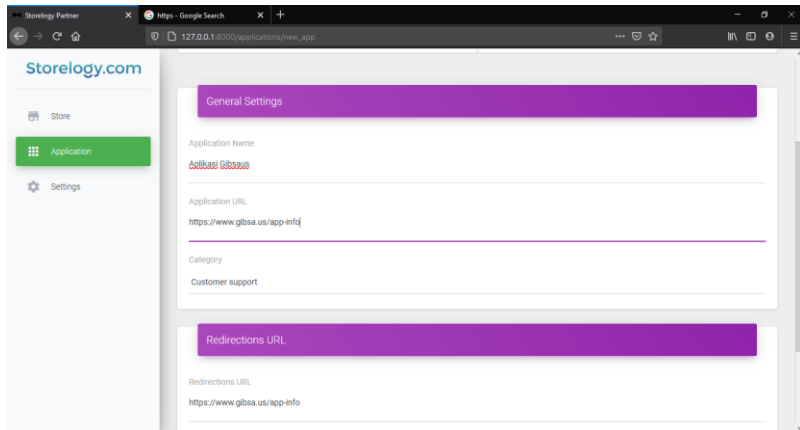
Hasil pengujian ditunjukkan pada Gambar 6.9, terlihat bahwa telah terbuat 3 buah toko baru yang bernama ‘toko bangunan’, ‘toko keramik’ dan ‘toko mobil’. Sehingga dapat disimpulkan bahwa pengujian terpenuhi.



Gambar 6.9 Hasil Pengujian Membuat Toko Baru

6.5 F-005 Mendaftarkan Aplikasi

Pengguna mendaftarkan aplikasi pihak ketiga baru, menggunakan informasi pribadi penulis, bernama ‘aplikasi gibsaus’ dan ‘aplikasi kedua’ tipe aplikasi public dan menggunakan *redirect link* bebas tapi sesuai seperti yang ditunjukkan pada gambar 6.10. Pengujian terpenuhi apabila kedua aplikasi telah terdaftar didalam basis data pada tabel *app_infos*.



Gambar 6.10 Pengujian Mendaftarkan Aplikasi

Gambar 6.11 adalah tampilan dari aplikasi *SQLYog* yang menunjukkan isi basis data pada tabel *app_infos*. Terlihat bahwa 2 entri terakhir adalah ‘aplikasi gibsaus’ dan ‘aplikasi kedua’ yang merupakan 2 aplikasi yang dibuat pada proses pengujian.

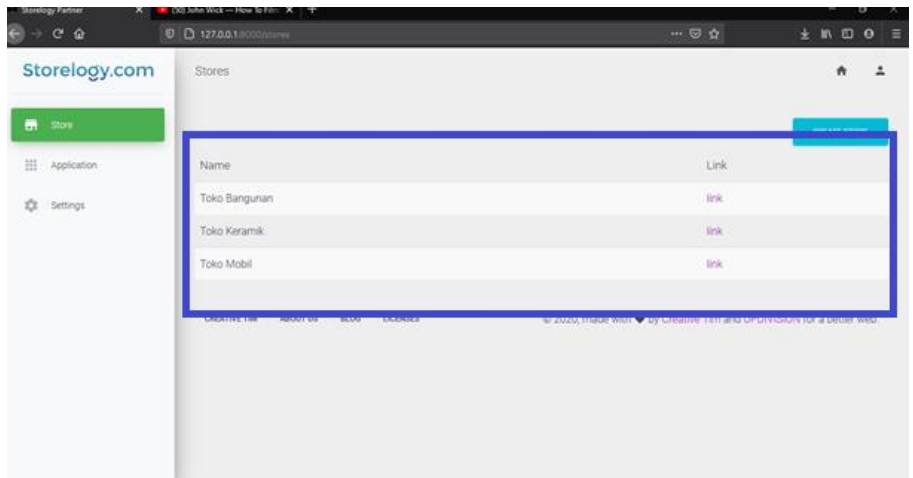
47	2 laswi57	2 http://wastuti.net/	(NULL)
48	1 aastuti	2 http://widodo.web.id/nihil-libero-corporis-excepturi-animi-est.html	(NULL)
49	2 leo36	2 http://www.wijayanti.org/qui-vel-aliquid-occaecat	(NULL)
50	9 mulyono20	2 https://puspasari.tv/est-assumenda-qui-placeat.html	(NULL)
54	1 good	1 http://127.0.0.1:8000/applications/new_app	(NULL)
56	1 Aplikasi	2 http://127.0.0.1:8000/applications/new_app	(NULL)
58	52 aplikasi gibsaus	2 http://127.0.0.1:8000/applications/new_app	(NULL)
59	52 Aplikasi Kedua	2 http://127.0.0.1:8000/applications/new_app	(NULL)
(data)	(NULL)	(NULL)	(NULL)

Dua Entri baru berhasil ditambahkan

Gambar 6.11 Hasil Pengujian Mendaftarkan Aplikasi

6.6 F-006 Melihat Toko yang Dibuat

Pengguna login menggunakan akun pribadi penulis (mhdfadlyhasan16a1@gmail.com) membuka rute `/stores/` Pengujian terpenuhi apabila toko-toko buatan pengguna ditampilkan

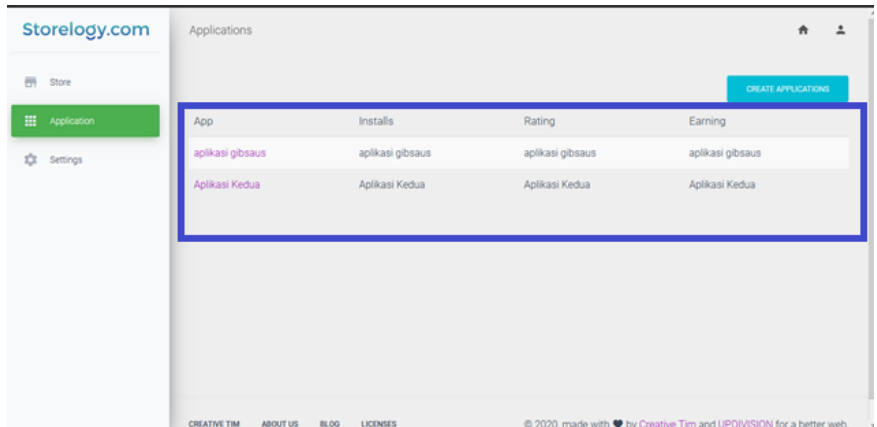


Gambar 6.12 Pengujian Melihat Toko yang Dibuat

Hasil pengujian ditunjukkan pada Gambar 6.12 diatas, terlihat bahwa tampilan ‘stores’ menampilkan tampilan store dan informasi 3 buah toko bernama ‘toko bangunan’, ‘toko keramik’ dan ‘toko mobil’ yang telah dibuat pada pengujian F-004. Sehingga dapat disimpulkan bahwa pengujian terpenuhi.

6.7 F-007 Melihat Aplikasi yang dikembangkan

Pengguna login menggunakan akun pribadi penulis (mhdfadlyhasan16a1@gmail.com) membuka rute /applications/. Pengujian terpenuhi apabila aplikasi-aplikasi pihak ketiga yang dikembangkan pengguna ditampilkan.

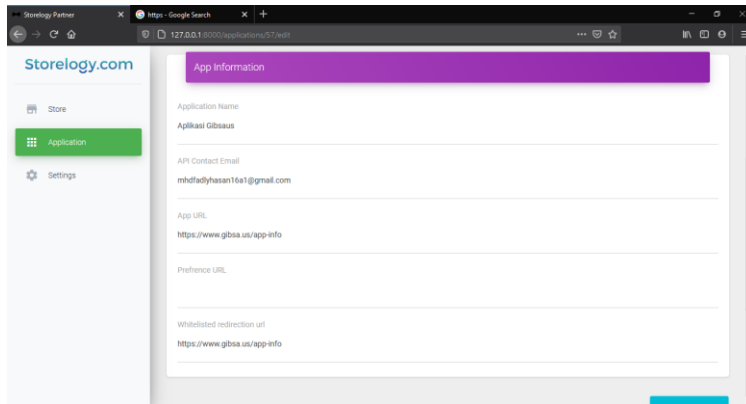


Gambar 6.13 Pengujian Melihat Aplikasi yang Dikembangkan

Gambar 6.13 adalah tampilan dari halaman di rute 'applications', terlihat ada 2 aplikasi yang ditampilkan dalam tabel, aplikasi ini adalah aplikasi yang dibuat pada pengujian F-005. Sehingga dapat disimpulkan bahwa pengujian terpenuhi.

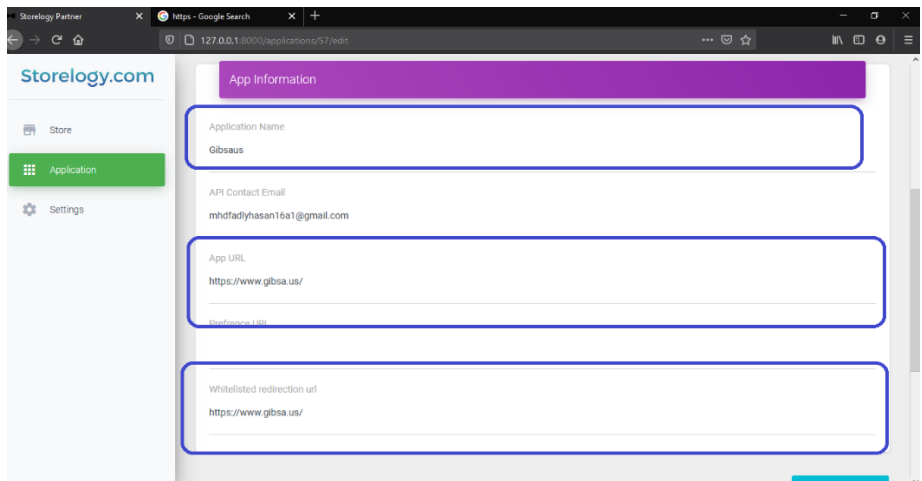
6.8 F-008 Mengubah detail aplikasi

Pengguna login dengan menggunakan akun yang ber-email mhdfadlyhasan16a1@gmail.com kemudian membuka aplikasi bernama 'aplikasi gibsaus', menekan link *settings*, kemudian mengubah nama aplikasi menjadi 'Gibsaus' dan mengubah informasi *website* dan *whitelisted redirection url* menjadi <https://gibsa.us> seperti yang ditunjukkan pada gambar 6.14. Pengujian terpenuhi apabila informasi nama, *website*, dan *whitelisted redirection url* berubah.



Gambar 6.14 Pengujian Mengubah Detail Aplikasi

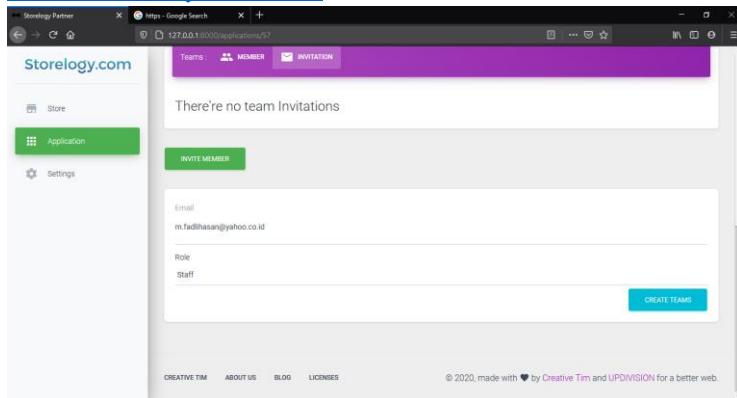
Gambar 6.15 adalah tampilan dari halaman di rute 'applications/edit' setelah pengujian yang menunjukkan detail informasi dari aplikasi yang dilakukan uji coba. Dapat terlihat bahwa informasi application name, app url, dan whitelisted redirection url berhasil diubah



Gambar 6.15 Hasil Pengujian Mengubah Detail Aplikasi

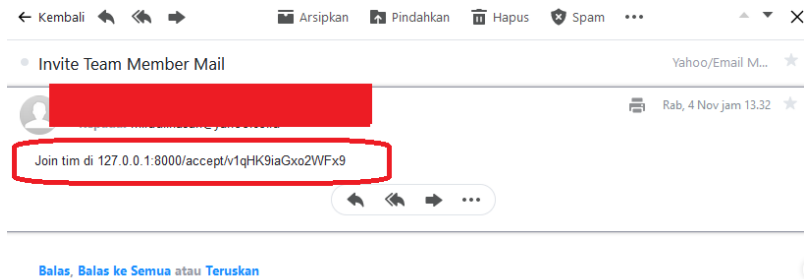
6.9 F-009 Menambahkan Anggota Tim Pengembang

Pengguna *login* dengan menggunakan akun mhdfadlyhasan16a1@gmail.com kemudian membuka aplikasi bernama ‘aplikasi gibsaus’ dan mengundang pengguna dengan email m.fadlihasan@yahoo.co.id seperti yang ditunjukkan pada gambar 6.16. Pengujian terpenuhi apabila email undangan dan *link* penerimaan undangan diterima oleh pengguna m.fadlihasan@yahoo.co.id.



Gambar 6.16 Pengujian Menambahkan Anggota Tim Pengembang

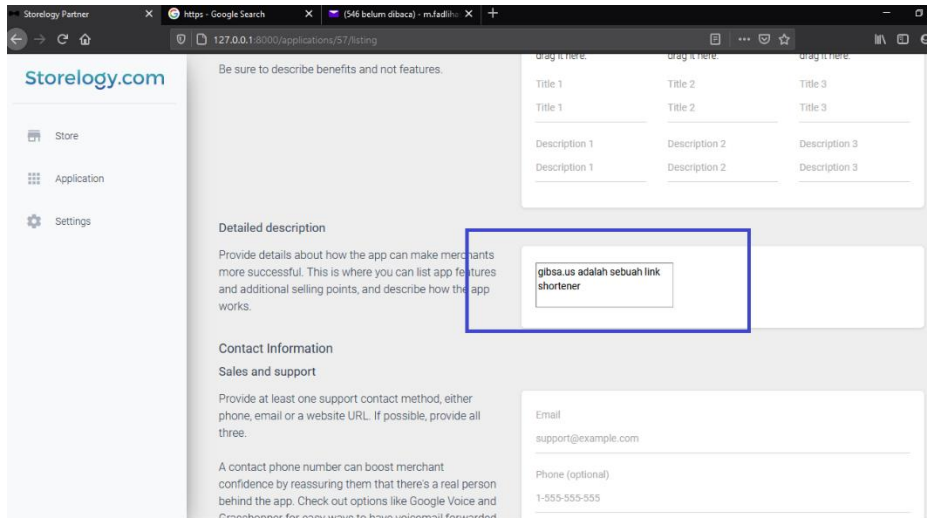
Gambar 6.17 adalah tampilan dari yahoo mail pengguna dengan email m.fadlihasan@yahoo.co.id, pesan email terbaru telah didapatkan dari email sistem yang berisi link penerimaan undangan anggota tim.



Gambar 6.17 Hasil Pengujian Menambahkan Anggota Tim Pengembang

6.10 F-010 Mengubah listing Aplikasi

Pengguna *login* dengan menggunakan akun mhdfadlyhasan16a1@gmail.com kemudian membuka aplikasi bernama ‘aplikasi gibsaus’, menekan link *manage listing*, dan mengubah informasi deskripsi menjadi “*gibsa.us* adalah sebuah link shortener”. Pengujian terpenuhi apabila informasi deskripsi berhasil diubah.

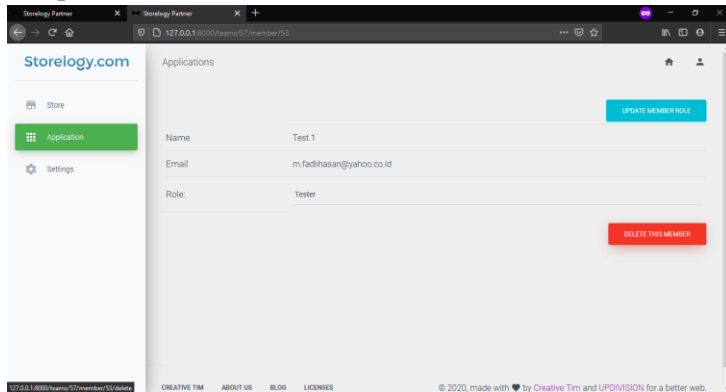


Gambar 6.18 Hasil Pengujian Mengubah Listing Aplikasi

Gambar 6.18 adalah tampilan dari halaman di rute 'applications/edit_listing' setelah informasi deskripsi diubah dan tombol submit ditekan. Dapat dilihat dalam *textarea* deskripsi bahwa informasi deskripsi berhasil diubah.

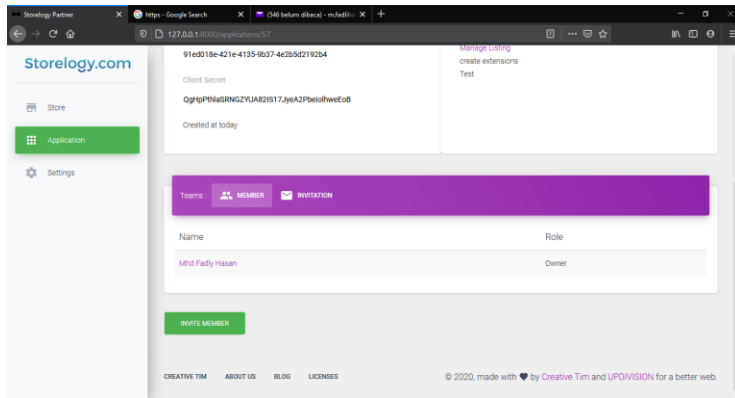
6.11 F-011 Menghapus Anggota Tim Pengembang

Pengguna *login* dengan menggunakan akun mhdfadlyhasan16a1@gmail.com kemudian membuka aplikasi bernama ‘gibsaus’, memilih anggota ‘test 1’ dan menekan tombol *delete this member* seperti yang ditunjukkan pada gambar 6.19. Pengujian terpenuhi apabila informasi keanggotaan ‘test 1’ berhasil dihapus.



Gambar 6.19 Pengujian Menghapus Anggota Tim Pengembang

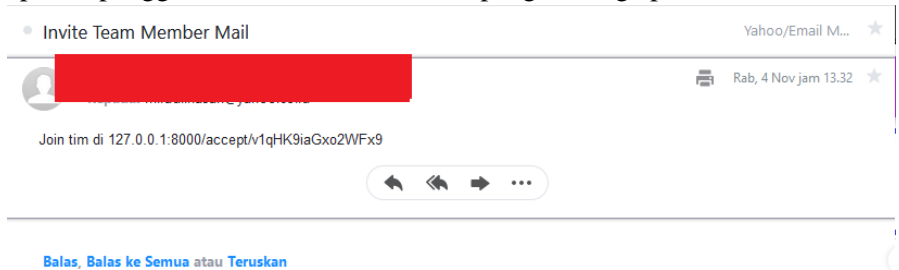
Gambar 6.20 adalah tampilan dari halaman ‘detailed_application’ yang menunjukkan aplikasi gibsaus, terlihat bahwa keanggotaan yang bernama test 1 berhasil dihapus.



Gambar 6.20 Hasil Pengujian Menghapus Anggota Tim Pengembang

6.12 F-012 Menerima Undangan Tim

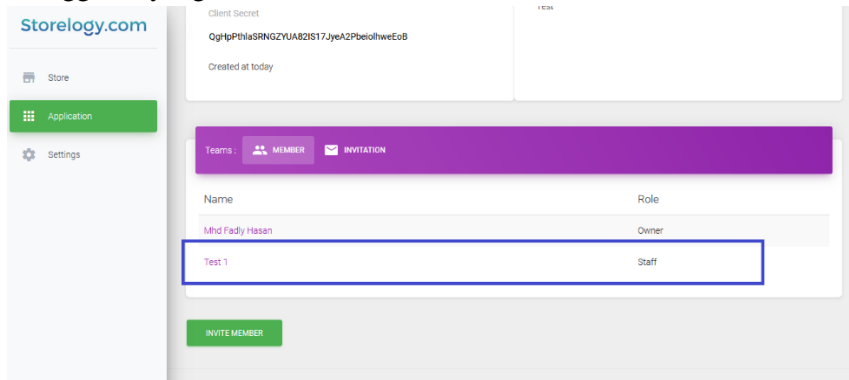
Pengguna yang telah *login* dengan email m.fadlihasan@yahoo.co.id dengan nama 'Test 1' membuka inbox pesan dan membuka email yang berisi *url* penerimaan anggota tim seperti yang ditunjukkan pada gambar 6.21. Pengujian terpenuhi apabila pengguna terdaftar kedalam tim pengembang aplikasi.



Gambar 6.21 Pengujian Menerima Undangan Tim

Gambar 6.22 adalah tampilan dari halaman 'detailed_aplication' setelah pengguna membukan email penerimaan keanggotaan, tampilan ini menunjukkan aplikasi gibsaus dan keanggotaan tim

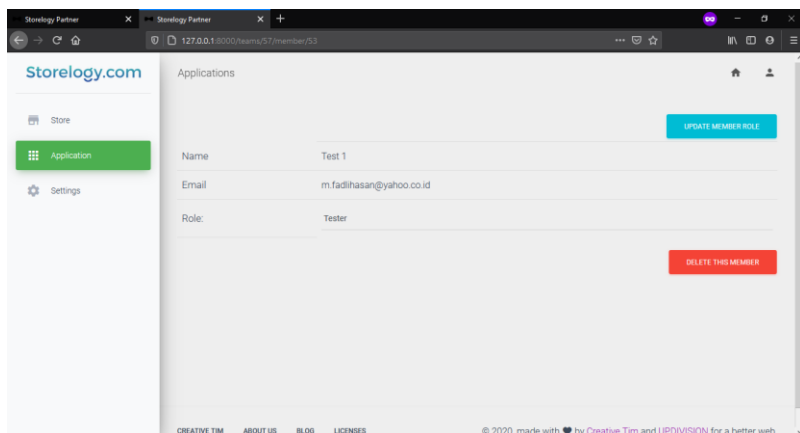
pengembangnya. Didalam bagian keanggotaan, terlihat bahwa keanggotaan yang bernama test 1 berhasil ditambahkan.



Gambar 6.22 Hasil Pengujian Menerima Undangan Tim

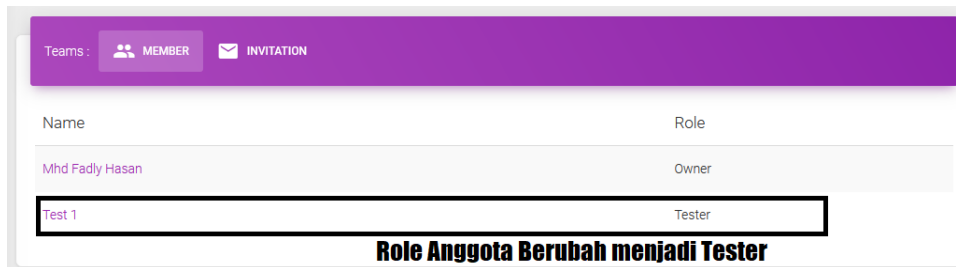
6.13 F-013 Mengubah Role anggota tim

Pengguna mengubah *role* anggota tim yang bernama 'test 1' menjadi 'tester' seperti yang ditunjukkan pada gambar 6.23. Pengujian terpenuhi apabila *role* 'test 1' berhasil diubah menjadi tester.



Gambar 6.23 Pengujian Mengubah Role anggota tim

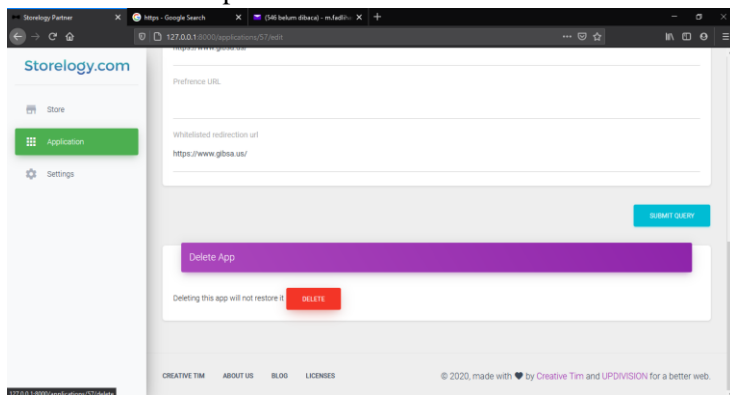
Gambar 6.24 adalah tampilan dari halaman ‘detailed_aplication’ setelah pengguna mengubah keanggotaan test 1 menjadi tester. Dapat dilihat pada bagian keanggotaan test 1 berhasil diubah menjadi tester.



Gambar 6.24 Hasil Pengujian Mengubah Role anggota tim

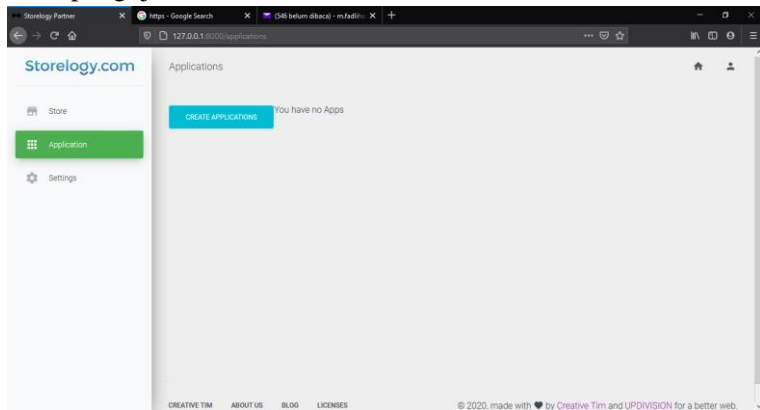
6.14 F-014 Menghapus Aplikasi yang Dikelola

Pengguna membuka aplikasi pihak ketiga ‘gibsaus’ dan menekan tombol *delete app* di halaman *settings* seperti yang ditunjukkan pada gambar 6.25. Pengujian terpenuhi apabila data aplikasi berhasil dihapus dari basis data.



Gambar 6.25 Pengujian Menghapus Aplikasi yang Dikelola

Gambar 6.26 adalah tampilan dari halaman ‘application’ setelah pengguna menghapus aplikasi ‘gibsaus’. Terlihat bahwa halaman application tidak menunjukkan aplikasi apapun karena aplikasi yang sekarang dikelola pengguna tidak ada. Sehingga dapat disimpulkan bahwa pengujian berhasil.



Gambar 6.26 Hasil Pengujian Menghapus Aplikasi yang Dikelola

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Kesimpulan yang didapat setelah melakukan pembuatan *Storelogy Partners* adalah sebagai berikut:

- *Storelogy Partners* memudahkan pengguna *Storelogy* untuk menampilkan dan mengubah informasi-informasi pengguna di sistem *storelogy* melalui satu *sub-sistem*.
- *Storelogy Partners* mampu membuat toko, aplikasi, dan akun pengguna baru untuk memudahkan pengguna *Storelogy*.
- *Storelogy Partners* dibangun menggunakan kerangka kerja *Laravel* dan menggunakan basis data *MYSQL*.
- *Storelogy Partner* yang telah dibangun telah memenuhi kebutuhan PT Picodio selama masa kerja praktik.

7.2 Saran

Saran dari penulis untuk pengembang selanjutnya sub-sistem *Storelogy Partners* adalah sebagai berikut:

- Pastikan kebutuhan sesuai dari hasil Analisa agar tidak menyebabkan kesalahan ketika diimplementasi.
- Lakukan pembuatan *client oauth* untuk aplikasi pihak ketiga baru di lapisan *back-end* sistem.
- Buatlah halaman 'home' yang menunjukkan data sesuai dengan sistem.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] "History of PHP," [Online]. Available: <https://www.php.net/manual/en/history.php.php>.
- [2] Rasmus Lerdorf, Kevin Tatroe, Peter MacIntyre, "What does PHP do?," in *Programming PHP*, O'Reilly Media, Inc., 2006, pp. 1-2.
- [3] "MySQL 8.0 Reference :: What is MySQL," Oracle, 8 January 2021. [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [Accessed 14 January 2021].
- [4] "jQuery," OpenJs Foundation, 2021. [Online]. Available: <https://jquery.com/>. [Accessed 14 January 2021].
- [5] "Bootstrap from Twitter," Twitter, 19 Agustus 2011. [Online]. Available: https://blog.twitter.com/developer/en_us/a/2011/bootstrap-twitter.html. [Accessed 10 January 2021].
- [6] M. Bean, "An Introduction to Laravel," in *Laravel 5 Essentials*, Birmingham, packt, 2011, pp. 1-6.

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Nama : Mhd Fadly Hasan
Tempat, Tanggal Lahir : Medan, 04 April 1998
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Asal : Jalan Pinang Baris Elok, No.1 Medan,
Medan-Sunggal, Sumatera Utara
Alamat Surabaya : JL.Wisma Permai II No.63 J8
Telepon : 082166353357
Email : mhdfadlyhasan16a1@gmail.com

PENDIDIKAN FORMAL

2017 – sekarang : Mahasiswa S1 Informatika ITS
2013 – 2016 : SMA Sultan Iskandar Muda Medan
2010 – 2013 : SMP Sultan Iskandar Muda Medan
2004 – 2010 : SD Sultan Iskandar Muda Medan

KEMAMPUAN

- *Web Programming* (HTML, PHP, Laravel, jQuery, Vue, Django)
- *Software Programming* (C, C++, Python, Java, Kotlin, Javascript)
- *Database Management* (MySQL, Firebase, Postgresql)
- *Software Perkantoran* (*Microsoft Office, Google Docs*)
- Bahasa (Indonesia, Inggris, Jepang)

AKADEMIS

Kuliah : Departemen Informatika, Fakultas Teknologi
Elektro dan Infromatika Cerdas, Institut
Teknologi Sepuluh Nopember Surabaya
Angkatan : 2017
Semester : 7 (Tujuh)
IPK : 3,54 (Semester 6)

[Halaman ini sengaja dikosongkan]